Sparse Representation Techniques for Hyperspectral Imaging

Naveed Akhtar

This thesis is presented for the degree of Doctor of Philosophy of The University of Western Australia School of Computer Science and Software Engineering. November, 2016
© Copyright 2016 by Naveed Akhtar
Dedicated to my father
Akhtar Ali Masoom
Abstract

Hyperspectral imaging quantizes scene radiance against several narrow wavelength ranges, thereby preserving the fine spectral details that are lost in conventional RGB cameras. This makes it an ideal imaging modality in numerous applications ranging from remote sensing to forensics and medical diagnostics. However, the low resolution of hyperspectral sensors, due to hardware constraints, hinders their pervasive use. In remote sensing, the low sensor resolution results in mixing the reflectances of materials, leading to errors in Earth exploration. In ground-based applications, the orders of magnitude difference between the contemporary RGB cameras and hyperspectral cameras makes the former preferable despite their inferior spectral characteristics.

Organized as a series of papers published by or submitted at prestigious venues in the fields of computer vision, remote sensing and artificial intelligence, this dissertation addresses the key problems in hyperspectral imaging to enable extensive use of this technology. It presents generic techniques and algorithms for (a) hyperspectral unmixing, (b) hyperspectral super-resolution and (c) image classification. The proposed techniques exploit the sparse representation framework that is grounded in optimization and non-parametric Bayesian theory. The proposed algorithms are systematically derived by capitalizing on the physical attributes of the signals that lead to state-of-the-art performance by the proposed techniques.

For hyperspectral unmixing, the proposed methods construct dictionaries from the libraries of pure material spectra provided by NASA and sparsely represent hyperspectral images over these dictionaries to separate their pixel constituents. The thesis presents two novel algorithms for sparse unmixing. The first algorithm, OMP-Star, makes the Orthogonal Matching Pursuit (OMP) strategy robust to the high mutual coherence of the spectral dictionaries. The second algorithm performs efficient Sparse Unmixing via Greedy Pursuit (SUnGP) by iteratively constructing and pruning the subspace of the likely constituents of the pixels. An effective technique for unsupervised unmixing is also proposed. It performs robust non-negative matrix factorization for unmixing, keeping in view the physical properties of the image.

Three different methods for hyperspectral super-resolution are proposed in this thesis. The first method extracts the constituent spectra of a hyperspectral image using dictionary learning. These spectra are transformed according to the spectral quantization of an RGB image, which is later represented over the transformed spectra. The resulting representations are combined with the original spectra to obtain a hyperspectral image with a resolution equal to that of the RGB image.
The non-parametric Bayesian framework is used for the first time for hyperspectral super-resolution in the second method, that also proposes a generic Bayesian sparse coding strategy to infer an ensemble of codes that are used to compute optimal super-resolution images. The third method incorporates the often-observed spatio-spectral smoothness of hyperspectral images into the Bayesian model with spatial and spectral kernels.

This thesis theoretically dismisses the misconception that using sparse representation for classification is inexpedient. It also proposes a generic discriminative Bayesian dictionary learning framework that shows state-of-the-art performance in various computer vision classification tasks. This framework is then extended to employ a joint representation model and to learn a classifier simultaneously along a Bayesian dictionary. This model is finally tailored to hyperspectral image classification by incorporating spectral smoothness into the basis with Gaussian Processes and exploiting contextual information in sparse coding. Effectiveness of the proposed method is shown on multiple real hyperspectral image datasets.
Acknowledgements

I thank Allah for enabling me to complete my thesis. I am grateful to my parents, especially my father who passed away during my PhD, for being an interminable source of inspiration for me. Their encouragement always gave me the strength to work hard and successfully achieve my goals. I am also grateful to my wife for her continuous support during my PhD, without which, my success would not have been possible. I also acknowledge the patience of my children Rahma and Arham that had let me focus on my work. I am grateful to my other family members as well for understanding my unavailability during the degree.

I would like to express my deepest gratitude to my supervisor Ajmal Mian. The high quality of my research originated from his guidance and continuous support throughout the degree. I am also grateful to my co-supervisor Faisal Shafait. It was only due to the invaluable and timely feedback of my supervisors on my work that I was able to continuously deliver during my PhD. I consider the skills that I learned from them as an asset for the rest of my research career. Along my supervisors, I am also grateful to my colleagues at the Machine Intelligence Group at UWA, especially Syed Zulqarnain Gilani for providing guidance and support in numerous matters that allowed me to focus on my research even better. I also thank all the researchers from my department who appreciated my work in seminars and personal meetings. Their appreciation kept me motivated for the high quality research.

In my PhD, I made use of public codes and databases. I am grateful to the researchers and their institutes for providing this data. These institutes include, Harvard University (Harvard hyperspectral database), Columbia University (CAVE database), California Institute of Technology (Caltech-101 database), Yale University (Extended Yale B) and University of Central Florida (UCF Sports action data), to name a few. I also acknowledge the providence of hyperspectral data from the Jet Propulsion Lab, NASA and US Geological Survey for my research. In the end, I thank the anonymous reviewers for providing positive and useful feedback on my work that resulted in improving the quality of my research articles.

This research was financially supported by The University of Western Australia under the Scholarship for International Research Fees (SIRF), UWA Top-Up scholarship and The Australian Research Council (ARC) under the Grant DP110102399. The Australian Government’s Research Training Scheme also financially contributed to this research after January 18, 2016.
## Contents

List of Tables vii  
List of Figures ix  
Publications Included in this Thesis xii  
Contribution of Candidate to Published Papers xiv

1 Introduction 1  
1.1 Applications .................................. 2  
1.2 Background and Definitions ...................... 3  
1.3 Thesis Overview ................................ 4  
1.3.1 Part I - Hyperspectral Unmixing .............. 5  
1.3.2 Part II - Hyperspectral Super-Resolution ..... 6  
1.3.3 Part III - Classification .................... 7  
1.4 Contributions ................................ 8

2 Repeated Constrained Sparse Coding with Partial Dictionaries for Hyperspectral Unmixing 12  
2.1 Introduction .................................. 12  
2.2 Problem Formulation ............................ 15  
2.2.1 Linear Mixing Model ......................... 15  
2.2.2 Hyperspectral Unmixing as Sparse Approximation .................. 15  
2.2.3 Mutual Coherence .......................... 16  
2.3 Proposed Solution ............................ 18  
2.3.1 Repeated Constrained Sparse Coding .......... 18  
2.3.2 Repeated Spectral Derivative ................ 20  
2.4 Results and Discussion ....................... 21  
2.4.1 Spectral Library .......................... 21  
2.4.2 Simulated Data ............................ 22  
2.4.3 Real Data ................................. 24  
2.5 Conclusion .................................. 26

3 Futuristic Greedy Approach to Sparse Unmixing of Hyperspectral Data 27  
3.1 Introduction ................................. 27
3.2 Hyperspectral Unmixing as Sparse Approximation . . . . . . . . . . . . 31
  3.2.1 Linear Mixing Model . . . . . . . . . . . . . . . . . . . . . . . 31
  3.2.2 Sparse Unmixing . . . . . . . . . . . . . . . . . . . . . . . . . 31
  3.2.3 Automatic Imposition of ASC with ANC . . . . . . . . . . . . . 33
3.3 Greedy Algorithms . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
3.4 Proposed Algorithms . . . . . . . . . . . . . . . . . . . . . . . . . . 36
  3.4.1 OMP-Star . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
  3.4.2 OMP-Star+ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
3.5 Coherence Reduction . . . . . . . . . . . . . . . . . . . . . . . . . . . 41
3.6 Experiments with Synthetic Data . . . . . . . . . . . . . . . . . . . . 44
  3.6.1 Data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44
  3.6.2 Analysis of Derivatives . . . . . . . . . . . . . . . . . . . . . . . 45
  3.6.3 Experiments for Endmembers Identification . . . . . . . . . . . . 47
  3.6.4 Experiments for Fractional Abundance . . . . . . . . . . . . . . 52
3.7 Experiments with Real Data . . . . . . . . . . . . . . . . . . . . . . . 55
3.8 Complexity Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
3.9 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 61

4 SUnGP: A Greedy Sparse Approximation Algorithm for Hyperspectral Unmixing 62
  4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62
  4.2 Unmixing as Sparse Approximation Problem . . . . . . . . . . . . . . 64
    4.2.1 Linear Mixing Model . . . . . . . . . . . . . . . . . . . . . . . 64
    4.2.2 Sparse Approximation of a Mixed Pixel . . . . . . . . . . . . . 65
  4.3 Greedy Sparse Approximation . . . . . . . . . . . . . . . . . . . . . . 66
  4.4 Proposed Algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . . 67
  4.5 Processing the Data for Greedy Algorithms . . . . . . . . . . . . . . . 69
  4.6 Experiments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70
    4.6.1 Results on Synthetic Data . . . . . . . . . . . . . . . . . . . . . 71
    4.6.2 Results on the Real-world Data . . . . . . . . . . . . . . . . . . 72
  4.7 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 74

5 RCMF: Robust Constrained Matrix Factorization for Hyperspectral Unmixing 75
  5.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75
  5.2 Problem formulation . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
    5.2.1 Linear mixing model . . . . . . . . . . . . . . . . . . . . . . . . 78
5.2.2 Unmixing as Constrained Matrix Factorization . . . . . . . . . . 79
5.3 Proposed Approach . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
  5.3.1 Objective function . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
  5.3.2 Optimization algorithm . . . . . . . . . . . . . . . . . . . . . . . . 82
5.4 Experiments with synthetic data . . . . . . . . . . . . . . . . . . . . . 87
  5.4.1 Data . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 87
  5.4.2 Evaluation metrics . . . . . . . . . . . . . . . . . . . . . . . . . . . . 88
  5.4.3 Benchmarking . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89
  5.4.4 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90
5.5 Experiments with real data . . . . . . . . . . . . . . . . . . . . . . . . 95
5.6 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 97

6 Sparse Spatio-spectral Representation for Hyperspectral Image Super-Resolution 99
6.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 99
6.2 Related Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 100
6.3 Problem Formulation . . . . . . . . . . . . . . . . . . . . . . . . . . . . 102
6.4 Proposed Solution . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 103
6.5 Experimental Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . 106
6.6 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 112
6.7 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 114

7 Bayesian Sparse Representation for Hyperspectral Image Super-Resolution 115
7.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 115
7.2 Related Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 117
7.3 Problem Formulation . . . . . . . . . . . . . . . . . . . . . . . . . . . . 118
7.4 Proposed Approach . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 119
  7.4.1 Bayesian Dictionary Learning . . . . . . . . . . . . . . . . . . . . 120
  7.4.2 Bayesian Sparse Coding . . . . . . . . . . . . . . . . . . . . . . . . 122
7.5 Experimental Evaluation . . . . . . . . . . . . . . . . . . . . . . . . . . 126
7.6 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 130
7.7 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 131

8 Hierarchical Beta Process with Gaussian Process prior for Hyperspectral Image Super-Resolution 132
8.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 132
10.5 Experiments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 188
  10.5.1 Extended YaleB . . . . . . . . . . . . . . . . . . . . . . . . . . 189
  10.5.2 AR Database . . . . . . . . . . . . . . . . . . . . . . . . . . . 191
  10.5.3 Caltech-101 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 192
  10.5.4 Fifteen Scene Category . . . . . . . . . . . . . . . . . . . . . . 195
  10.5.5 UCF Sports Action . . . . . . . . . . . . . . . . . . . . . . . . 196
10.6 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 196
10.7 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 198

11 Joint Bayesian Dictionary and Classifier Learning 200
  11.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 200
  11.2 Problem Settings and Preliminaries . . . . . . . . . . . . . . . . . . 201
  11.3 Proposed Approach . . . . . . . . . . . . . . . . . . . . . . . . . . . 203
    11.3.1 The Model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 203
    11.3.2 Inference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 205
  11.4 Experiments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 208
    11.4.1 Face recognition . . . . . . . . . . . . . . . . . . . . . . . . . . 209
    11.4.2 Object classification . . . . . . . . . . . . . . . . . . . . . . . 211
    11.4.3 Scene categorization . . . . . . . . . . . . . . . . . . . . . . . 211
    11.4.4 Action recognition . . . . . . . . . . . . . . . . . . . . . . . . . 212
  11.5 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 213
  11.6 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 215

12 Non-parametric Coupled Bayesian Dictionary and Classifier Learning for Hyperspectral Classification 216
  12.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 217
  12.2 Preliminaries and Background . . . . . . . . . . . . . . . . . . . . . 220
  12.3 Proposed Approach . . . . . . . . . . . . . . . . . . . . . . . . . . . 222
    12.3.1 The Model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 222
    12.3.2 Inference . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 224
    12.3.3 Dictionary Size . . . . . . . . . . . . . . . . . . . . . . . . . . . 228
    12.3.4 Classification . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 229
  12.4 Experiments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 229
    12.4.1 Indian Pines Image . . . . . . . . . . . . . . . . . . . . . . . . . 231
    12.4.2 Salinas Image . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 234
    12.4.3 Pavia University Image . . . . . . . . . . . . . . . . . . . . . . . 236
  12.5 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 238
12.6 Conclusion ..................................................... 240

13 Conclusion ........................................................................... 241

A Appendix A ............................................................................. 244

B Appendix B ............................................................................. 245

B.1 Derivation of the Gibbs Sampling Equations ....................... 245

B.1.1 The Model ....................................................................... 245

B.1.2 Gibbs Sampling Equations ............................................... 245

B.2 MSE Analysis for Lemma 1 ................................................... 248

C Appendix C ............................................................................. 250

C.1 Further Ground-based Image Results ................................. 250

C.2 Further AVIRIS Image Results ............................................. 254

C.3 Robustness to Misalignment ............................................... 256

C.4 Illustration of Blur in CAVE Image ..................................... 256

D Appendix D ............................................................................. 257

D.1 Dictionary Visualization ..................................................... 257

D.2 Gibbs Sampling Analysis .................................................... 257

D.3 Performance Dependence on Training Data Size ................ 260

E Appendix E ............................................................................. 262

E.1 Graphical Representation ................................................... 262

E.2 Joint Probability Distribution .............................................. 262

E.3 Gibbs Sampling Equations .................................................. 262

Bibliography .................................................................................. 264
List of Tables

3.1 Comparison of greedy algorithms for fractional abundance estimation 56
3.2 Experiments of Table 3.1, repeated with correlated noise ............... 56
3.3 Processing time for unmixing by greedy pursuit algorithms ............. 60
4.1 Processing time of greedy algorithms for unmixing ...................... 72
5.1 Average spectral angle of resulting spectra ............................. 91
5.2 RMSE of reconstructed abundance matrices ............................. 91
5.3 Mean computation time for unmixing one thousand pixels .............. 94
6.1 Benchmarking of the approach on CAVE database ....................... 110
6.2 Benchmarking on Harvard database ................................... 110
7.1 Benchmarking of the proposed approach ................................ 127
7.2 Exhaustive experiment results ........................................... 128
8.1 Benchmarking on remote sensing images ................................ 144
8.2 Benchmarking on ground-based images ................................ 146
9.1 Recognition accuracies on the AR database ............................. 164
9.2 Performance gain with SA-CRC classification .......................... 164
9.3 Recognition accuracies on Extended YaleB ............................. 166
9.4 Classification accuracies on the Caltech-101 dataset .................. 167
9.5 Computation time for classification on Caltech-101 dataset ......... 167
9.6 Classification accuracies on UCF Sports Action dataset ............... 168
10.1 Recognition accuracy on the Extended YaleB .......................... 190
10.2 Recognition accuracy on the AR database ............................. 192
10.3 Classification results on the Caltech-101 dataset ..................... 194
10.4 Computation time for training and testing on Caltech-101 .......... 194
10.5 Classification accuracy on Fifteen Scene Category dataset .......... 195
10.6 Classification rates on UCF Sports Action data ....................... 197
11.1 Face recognition results on Extended YaleB database [77]. Results
are averaged over ten experiments. The time is given for classifying
a single test sample. .................................................. 209
11.2 Face recognition on AR database [140]. Results are averaged over ten
experiments. The time is for a single test sample. ....................... 211
11.3 Object classification on Caltech-101 [74].

11.4 Classification accuracies (%) on Fifteen Scene Category dataset [121] using Spatial Pyramid Features. The time for computing a single test sample is given in milli-seconds.

11.5 Action recognition on UCF Sports Action database [172].

12.1 Classification accuracy (%) for the Indian Pines Image.

12.2 Classification accuracy (%) for the Salinas Image.

12.3 Classification accuracy (%) for the Pavia University Image.

C.1 Further results on Harvard database.

C.2 Results on CAVE database.

D.1 Potential Scale Reduction Factors for key parameters.
List of Figures

2.1 Illustration of hyperspectral image cube .................................. 13
2.2 Comparison of coherence matrices with and without spectral derivatives 17
2.3 Illustration of coherence reduction with band removal ...................... 18
2.4 Repeated Constraint Sparse Coding algorithm .............................. 19
2.5 Illustration of computing multiple sparse codes with RCSC ................. 20
2.6 Repeated Spectral Derivative algorithm ................................... 21
2.7 Comparison of results of RCSC, RSD, CSC and DLR [19] .................. 23
2.8 Comparison of results using Eq. 2.6 and Eq. 2.7 .......................... 24
2.9 Results of RCSC and RSD on real hyperspectral data ....................... 25

3.1 Illustration of hyperspectral data cube ................................... 28
3.2 Analysis of derivatives ....................................................... 46
3.3 Example of white and correlated noise .................................... 47
3.4 Smoothing effect in derivatives ............................................. 48
3.5 Parameter selection of OMP-Star ......................................... 49
3.6 Performance of greedy algorithms on hyperspectral unmixing .......... 51
3.7 Unmixing fidelity as a function of SNR ................................ 52
3.8 Error in the computed fractional abundances .............................. 53
3.9 Examples of estimated fractional abundances by OMP-Star+ ............. 55
3.10 Mineral map of Cuprite mining district Nevada ............................ 58
3.11 Qualitative comparison of fractional abundance maps ................. 59
3.12 Example of typical iterations of OMP-Star ............................... 60

4.1 Illustration of a hyperspectral image cube .................................. 63
4.2 Comparison of SUnGP results on synthetic data .......................... 73
4.3 Comparison of SUnGP results on the real data .......................... 74

5.1 Hyperspectral mixing illustration .......................................... 76
5.2 Illustration of additive correlated noise .................................. 88
5.3 Examples of the endmembers recovered by the approach ................. 92
5.4 Average spectral angle as a function of the number of endmembers .... 92
5.5 RMSE as a function of the number of endmembers ....................... 93
5.6 Average spectral angle as a function of SNR ........................... 94
5.7 RMSE as a function of SNR ............................................... 94
5.8 Fractional abundance analysis of the AVIRIS data ....................... 96
5.9 Extracted endmembers from the AVIRIS image ........................... 97
11.2 (a) Dictionary size as a function of Gibbs Sampling iterations for Extended YaleB. The first 100 iterations are shown for different values of $a_o$ and $b_o$. (b) Worst-case Point Scale Reduction Factor (PSRF) [75] for $\pi_k^c, \forall c, \forall k$ as a function of the sampling iterations for Extended YaleB.

12.1 Graphical representation of the Bayesian model.

12.2 Sample band images for the scenes.

12.3 Comparison of the dictionary atoms.

12.4 Classification maps for the Indian Pines image.

12.5 Classification maps for the Salinas image.

12.6 Classification maps for the Pavia University image.

12.7 Dictionary size reduction with the Gibbs sampling iterations.

C.1 Illustration of robustness of approach to image misalignment.

D.1 Visualization of the most active dictionary atoms.

D.2 Visual inspection of mixing for the AR database.

D.3 Visual inspection of mixing for Fifteen Scene Category.

D.4 Worst-case PSRF as a function of Gibbs sampling iterations.

D.5 Classification accuracy as a function of training data size.

E.1 Graphical representation.
Publications Presented in the Thesis

This thesis is a compilation of papers accepted by and submitted to prestigious fully refereed international journals and conferences. The bibliographical details of these papers are outlined below. Each paper has its own significant contribution and forms a thesis chapter. The chapter number for each paper is also mentioned.

**Journal Articles**


(Chapter 10)


(Chapter 3)


(Chapter 5)


(Chapter 9)


(Chapter 12)

**Conference Papers**


(Chapter 7)


Note: TPAMI (IF 6.07) is ranked 1st in Computer Vision among 353 titles in this area according to the SJR ranking. Google Scholar ranks CVPR as the highest impact venue for Computer Vision research. ECCV has the 6th highest impact among all Computer Vision journals and conferences according to Google Scholar, that also ranks TGRS (IF 3.32) as the 2nd highest impact venue in Remote Sensing. TNNLS (IF 4.85) is ranked 8th among 140 journals in Artificial Intelligence by SJR. PR (IF 3.39) has the 7th highest impact among all Computer Vision journals and conferences according to Google Scholar. The mentioned impact factors and rankings are for year 2015/16.

All the above mentioned accepted conference papers were unanimously recommended for acceptance by the reviewers in the preliminary evaluation of the review process. Moreover, [6] was also recognized as the second best extreme imaging research in AustralAsia in the fields of engineering, computer science, physical sciences, medicine, astronomy and archaeology, by Canon Information Systems Research Australia (CISRA).
Contribution of Candidate to Published Work

My contribution in all the papers was 85%. I designed and implemented the algorithms, performed the experiments and wrote the papers. My supervisors reviewed the papers and provided useful feedback for improvement.
Spectral response of objects generally contains vital information regarding their material composition. However, contemporary computer vision techniques are unable to tap into this information due to the gross radiance quantization performed by the RGB cameras. In this regard, hyperspectral imaging can revolutionize computer vision research because it is able to preserve the fine spectral details of a scene. It integrates scene radiance under multiple basis functions to acquire a high-fidelity spectral response. Various computer vision, forensics, remote sensing and medical diagnostic tasks have recently been reported to benefit from the spectral properties of hyperspectral images.

Whereas spectral characteristics of hyperspectral imaging are desirable in multiple applications, the low spatial resolution of hyperspectral cameras currently hinders their ubiquitous use. The low resolution results from hardware constraints and it is not straightforward to improve it for hyperspectral imaging. On the other hand, RGB cameras are able to achieve resolution that is orders of magnitude better than their hyperspectral counterparts. This makes them a default choice in many applications despite their inferior spectral properties. Such applications can benefit greatly from hyperspectral imaging if the resolution of hyperspectral images can be enhanced to the level of RGB cameras.

Due to the low image resolution, multiple materials can be present in the area corresponding to a single pixel of a hyperspectral image, especially in remote sensing images that are taken several kilometers away from the Earth’s surface. This often makes a hyperspectral pixel a mixture of multiple spectra that belong to different materials. To fully exploit hyperspectral imaging, it is imperative to develop methods that accurately un-mix the spectral signatures into their constituents. Moreover, effective methods that can classify material spectra with high confidence are also highly desirable for the widespread use of hyperspectral imaging technology.

This thesis addresses three key problems in hyperspectral imaging to enable pervasive use of this technology in computer vision and other research areas. These problems include, (1) unmixing of the hyperspectral pixels into their constituent spectra and computing their proportions in the mixtures, (2) enhancing the resolution of hyperspectral images, i.e. hyperspectral super-resolution, to the level of RGB images; and (3) classification of the spectral signatures of materials. Exploiting the sparse representation frame work for image analysis, this dissertation develops state-of-the-art techniques to address these problems.
Chapter 1. Introduction

1.1 Applications

The emerging modality of hyperspectral imaging is proving beneficial in many applications. It has also demonstrated performance gain over conventional imaging for the key computer vision tasks of recognition, classification and segmentation. Nevertheless, the unavoidable inferior resolution of the images is currently a bottleneck in pervasive use of this modality. By enhancing hyperspectral image resolution and performing effective spectral unmixing and classification, the techniques developed in this thesis find applications in various research areas. The list of the applications is extensive and it is not possible to enumerate them all. Nevertheless, few broad application areas along the possible directions of exploiting the developed techniques are briefly discussed below.

- **Earth exploration** benefits from hyperspectral imaging by measuring the spectral reflectance of the Earth’s surface with imaging spectrometers. The methods of hyperspectral unmixing developed in the thesis are directly related to this area of research. Accurate unmixing of hyperspectral pixels can result in precise knowledge about the mineral composition of the Earth’s surface. Moreover, the developed super-resolution techniques can also be used for sharp hyperspectral remote sensing images. Many current hyperspectral remote sensing platforms are also installed with multispectral, RGB or panchromatic imaging systems. The image fusion based hyperspectral super-resolution techniques presented in the dissertation can be readily exploited by such configurations.

- **Document analysis** can benefit from hyperspectral imaging by analyzing the spectral response of the documents. For instance, differentiating between fake and real documents based on the spectra of the used inks can be performed. The developed unmixing techniques can be used to find the ink compositions for a reliable analysis.

- **Medical diagnostic** often requires to study composition of a substance. High resolution hyperspectral imaging and an ability to accurately classify the spectral response of a material can find numerous applications in this area of research. For instance, precise spectral response of affected skin and accurate classification of spectra may non-invasively diagnose melanoma. The techniques developed in this thesis provide the basic ingredients for effective use of hyperspectral imaging in medical diagnostics.

- **Robotics** can directly benefit from the classification techniques presented in
the dissertation. Recognition and classification are long standing problems in robotics. The generic classification techniques proposed in the thesis can be readily used in robotics.

- **Surveillance and tracking** are generally performed with remote sensing instruments in military applications. High resolution hyperspectral imaging with such instruments can improve performance by exploiting the spectral information.

Along the above mentioned applications, various other computer vision tasks can also take advantage of high resolution hyperspectral imaging. These tasks include, but are not limited to, human face recognition, human action recognition, pedestrian detection, object classification and scene understanding.

### 1.2 Background and Definitions

Each chapter of the thesis provides its relevant background and definitions of the used terms. The technical terms used in the remainder of this chapter are informally described here, along a brief description of the sparse representation framework. The description is intended to help in contextual understanding of the thesis contributions in the sparse representation literature.

Consider a high-dimensional signal $x \in \mathbb{R}^L$ that can be expanded as $x = \Phi \alpha$. In the sparse representation literature, the basis $\Phi \in \mathbb{R}^{L \times K}$ is known as the **dictionary**, whereas the vector $\alpha \in \mathbb{R}^K$ is often called the **representation** of the signal over the dictionary. The dictionary is generally considered to be an over-complete set of the basis vectors (dictionary atoms), therefore, many coefficients in $\alpha$ are expected to be zero or extremely small, i.e. $\alpha$ is **sparse**. Hence, the representation vectors are also known as the sparse codes of the signals. Developing techniques that learn suitable sparse codes and/or dictionaries for various computer-vision/signal-processing tasks is the main topic of interest in the sparse representation literature.

Given that a dictionary is known beforehand, sparse coding methods focus on high-fidelity signal reconstruction by solving optimization problems that encourage sparse representations. Sparsity is mostly induced in the representations by restricting their $\ell_0$ or $\ell_1$-norms. The $\ell_0$-norm (pseudo) counts the number of the non-zero coefficients in $\alpha$, whereas the $\ell_1$-norm computes the absolute sum of $\alpha$’s coefficients. The sparse coding algorithms that restrict the $\ell_0$-norm are collectively known as greedy pursuit algorithms. These algorithm are computationally efficient, however they are often less accurate than those restricting the $\ell_1$-norm of the repre-
sentations. This trade-off makes both $\ell_0$ and $\ell_1$-norm based sparsity interesting for hyperspectral image analysis.

A wide range of dictionary learning techniques are also available that solve optimization problems to additionally learn dictionaries along the sparse codes. Using optimization constraints, the learned dictionaries are often made suitable for different tasks. Note that, by computing both the dictionary and the sparse codes we factorize the training signals into two complementary matrices. Therefore, dictionary learning algorithms are also seen as matrix factorization techniques.

The algorithms for solving optimization problems learn point estimates of the dictionaries and the sparse codes. They are also not well-suited for incorporating prior domain knowledge into the representations in a principled manner. In this regard, Bayesian framework is the most suitable choice. Bayesian dictionary learning computes posterior probability distributions over the dictionary atoms, thereby learning an ensemble of dictionaries that also take advantage of the domain knowledge through suitable priors over the dictionary atoms. Moreover, Bayesian dictionaries can be learned non-parametrically, in the sense that the key parameter of ‘dictionary-size’ is inferred by the learning process itself.

1.3 Thesis Overview

This thesis is organized as a series of papers published by or submitted to prestigious journals and conferences in the fields of Computer Vision, Remote Sensing and Artificial Intelligence. The thesis is divided into three major parts that respectively deal with the problems of hyperspectral unmixing, hyperspectral super-resolution and classification. Each chapter of the thesis is composed of an independent research paper that contributes to the coherent story of its relevant part. Combined, the three parts address the three fundamental problems in hyperspectral imaging by proposing a wide range of algorithms under the sparse representation framework. The proposed algorithms follow the natural order of optimization-based sparse coding to dictionary learning, to Bayesian dictionary learning.

In a bird’s eye view, the first part of the thesis opens by studying the $\ell_1$-norm based sparse coding for hyperspectral unmixing in Chapter 2. Then, it develops two computationally efficient greedy pursuit algorithms in Chapters 3 and 4 that show unmixing accuracies comparable to that achieved through $\ell_1$-norm minimization. Sparse coding techniques assume the desired dictionary to be known beforehand. Chapter 5 removes this assumption for hyperspectral unmixing and develops a robust matrix factorization algorithm.
1.3. Thesis Overview

The second part of the thesis deals with the problem of hyperspectral super-resolution. Chapter 6 proposes an image fusion framework to combine spatial information in RGB images with hyperspectral images using dictionary learning and a greedy pursuit algorithm. The proposed algorithm also incorporates contextual information of images in the sparse codes. In Chapter 7, the framework is translated into Bayesian settings. Chapter 8 exploits the relative smoothness of spectra in Bayesian dictionary learning with Gaussian Processes [167] and uses a kernel to incorporate spatial information in the Bayesian sparse codes.

Sparse representation based classification is the topic of the third part of the thesis. After a theoretical argument in favor of using sparse representation for classification in Chapter 9, a generic discriminative Bayesian dictionary learning approach is developed in Chapter 10. This scheme is enhanced to learn a linear classifier along the dictionary in Chapter 11. Finally, it is customized for hyperspectral classification by incorporating the relative smoothness of the spectral signals into the learned basis in Chapter 12. A comprehensive overview of each chapter of the dissertation is provided below.

1.3.1 Part I - Hyperspectral Unmixing

Chapter 2: *Repeated constrained sparse coding with partial dictionaries for hyperspectral unmixing.*

This chapter develops two different methods for hyperspectral unmixing that minimize the $\ell_1$-norm of pixel representations over a dictionary composed of reflectances of pure materials. The first method, Repeated Constrained Sparse Coding (RCSC) shows an overall robustness against noise, whereas the second method, Repeated Spectral Derivative (RSD) shows better performance at high signal to noise ratio. While repeatedly solving the sparse coding problem for hyperspectral unmixing, RCSC modifies the available dictionary by systematically ignoring few of its spectral channels in each run. This results in coherence reduction of the dictionary that improves the unmixing accuracy when all the sparse codes are jointly used for unmixing. The RSD method uses spectral derivatives for coherence reduction, however, while repeatedly solving the sparse coding problem, the derivatives are not operated on few spectral channels to make the operation less sensitive to noise.

Chapter 3: *Futuristic greedy approach to sparse unmixing of hyperspectral data.*

A greedy pursuit algorithm is developed for hyperspectral unmixing in this chapter. The algorithm mitigates the local optimality issue of the Orthogonal Matching
Pursuit [33] strategy by deciding its current iteration based on the possible future iterations. This makes it suitable for the dictionaries with high mutual coherence [196] - a common property of spectral dictionaries [102]. An enhancement of the algorithm is proposed to incorporate the physical constraints over hyperspectral signals into the sparse codes. It is demonstrated that the resulting algorithm achieves unmixing accuracy comparable to that obtained by the $\ell_1$-norm minimization of the sparse codes, but in less than half the time. This chapter also presents an important theoretical finding that $\ell_1$-normalization of spectral dictionary atoms automatically imposes the well-known Abundance Sum-to-one Constraint (ASC) [25] in unmixing.

Chapter 4: *SUnGP: A greedy sparse approximation algorithm for hyperspectral unmixing.*

In contrast to the futuristic approach in Chapter 3, this chapter introduces a subspace pruning technique for sparse unmixing under the greedy pursuit strategy that remains robust against the high coherence of the spectral dictionaries. The proposed Sparse Unmixing via Greedy Pursuit (SUnGP) algorithm imposes non-negativity on the sparse codes to enforce Abundance Non-negativity Constraint (ANC) [25] in unmixing. The overall approach takes advantage of the theoretical findings in Chapter 3 and the spectral derivatives [200] used in Chapter 2 to achieve unmixing results comparable to those of $\ell_1$-norm based sparse unmixing, albeit nearly six times faster.

Chapter 5: *RCMF: Robust constrained matrix factorization for hyperspectral unmixing.*

Appropriate spectral dictionaries may not always be available for hyperspectral unmixing. Therefore, this chapter removes the need of pre-defined spectral dictionaries by learning a dictionary of material reflectances from the image itself. The dictionary is computed by a matrix factorization algorithm that is guided by the physical attributes of hyperspectral images. The proposed algorithm is made robust to outliers to perform accurate unmixing in a completely unsupervised manner.

1.3.2 Part II - Hyperspectral Super-Resolution

Chapter 6: *Sparse spatio-spectral representation for hyperspectral image super-resolution.*

This chapter presents a framework to improve the resolution of hyperspectral images up to the level of contemporary RGB cameras. The proposed framework separates the distinct spectra in a low resolution hyperspectral image using dictionary learn-
ing. Then, these spectra are transformed according to the spectral quantization of an RGB image of the same scene. The RGB image is represented over the transformed spectra using a greedy pursuit algorithm. The proposed algorithm embeds the spatial information in the RGB image into the sparse codes, which are later used with the original spectra to construct the super-resolution hyperspectral image.

Chapter 7: Bayesian sparse representation for hyperspectral image super-resolution

Non-parametric Bayesian theory is used for the first time for hyperspectral super-resolution in this chapter. The framework proposed in Chapter 6 is translated into Bayesian settings by using Beta Process [157] based dictionary learning and proposing a novel Bayesian sparse coding scheme. The proposed scheme is theoretically analyzed for optimal sparse representations and the results are exploited in achieving highly accurate hyperspectral super-resolution images.

Chapter 8: Hierarchical Beta Process with Gaussian Process prior for hyperspectral image super-resolution.

Generally, the reflectances of materials are relatively smooth functions of spectral wavelength [216]. Moreover, scene patterns often vary slowly along the spatial dimensions, especially in remotely sensed images. The approach in Chapter 7 is extended for such cases in this chapter. A non-parametric Bayesian dictionary learning model is proposed that places Gaussian Process [167] priors over the dictionary atoms to incorporate spectral smoothness into the inferred basis. The hierarchical model also introduces a spatial kernel in Bayesian settings to exploit the contextual information in the images. Experiments show that the proposed model is especially suitable for remote sensing images.

1.3.3 Part III - Classification

Chapter 9: Efficient classification with sparsity augmented collaborative representation.

It has recently been argued that it is the collaboration and not the sparseness of representations that results in accurate sparse representation based classification [233]. This chapter theoretically analyzes this argument and finds that contrary to the original claim, a representation’s sparseness plays an explicit role in its accurate classification. Hence, instead of ignoring sparsity for computational gains, dense representations are augmented with greedily computed sparse representations and an efficient classification criterion is proposed for the resulting representation. The
proposed approach is more accurate than the classical sparse representation based
classification scheme [213], at the same time, it is computationally more efficient
than the approaches ignoring the sparsity altogether.

Chapter 10: *Discriminative Bayesian dictionary learning for classification.*
A generic discriminative Bayesian dictionary learning technique is developed in this
chapter that computes the dictionaries non-parametrically. The proposed learning
model induces discrimination in a dictionary by associating its atoms with different
classes of the training data under different sets of Bernoulli distributions. Gibbs
sampling [62] equations are derived for Bayesian inference over the proposed model
and its non-parametric nature is analyzed. The developed technique has been eval-
uated for the tasks of face, object, scene and action recognition to demonstrate its
effectiveness.

Chapter 11: *Joint Bayesian dictionary and classifier learning.*
A technique to jointly learn a linear classifier along a discriminative Bayesian dic-
tionary is developed in this chapter. The proposed non-parametric Bayesian model
allows simultaneous inference over the dictionary and the classifier, which customizes
the classifier to be used in conjunction with the inferred discriminative dictionary.
The natural coupling between the dictionary and the classifier results in accurate
and efficient sparse representation based classification.

Chapter 12: *Non-parametric coupled Bayesian dictionary and classifier learning
for hyperspectral classification.*
The model developed in Chapter 11 is tailored to hyperspectral classification in
this chapter. The resulting model uses Gaussian Processes [167] with kernels that
promote high correlation between the nearby coefficients of the dictionary atoms,
that are used to represent relatively smooth spectra. An effective novel criterion is
also proposed to predict the class label of the test hyperspectral pixels. The criterion
takes advantage of the contextual information in the imaged scene.

1.4 Contributions

The major contributions of the thesis are summarized below. The contributions
sequenced as they appear in the thesis.

- Two novel methods for sparse unmixing of hyperspectral data using $\ell_1$-norm
  minimization of the representations are proposed. One method shows overall
robustness to noise, whereas the other shows good performance at high signal to noise ratio.

- Two novel greedy pursuit algorithms are developed that show robustness to high mutual coherence of the dictionaries. The algorithms are generic in their nature, nevertheless they show improved performance for sparse unmixing by also incorporating the physical constraints of the spectral signals.

- An important theoretical result is presented for hyperspectral unmixing, showing that the well-known Abundance Sum-to-one Constraint [25] is automatically imposed in sparse unmixing by $\ell_1$-normalization of the dictionary atoms.

- The use of spectral derivatives [200] with greedy pursuit algorithms is introduced for the first time in hyperspectral unmixing.

- A robust matrix factorization algorithm is proposed for unsupervised unmixing of hyperspectral images. The algorithm computes the basis vectors for image representations as combinations of the pixels themselves, thereby attaching physical significance to each basis vector.

- A sparse representation based hyperspectral super-resolution framework is introduced and a greedy pursuit algorithm is proposed for the framework that exploits physical properties of hyperspectral images for improved performance.

- A generic Bayesian sparse coding scheme is developed that can be used with Bayesian dictionaries. Theoretical analysis is provided to show optimality of the sparse codes computed under the proposed scheme.

- A non-parametric Bayesian sparse representation model is proposed that uses Gaussian Processes [167] as the base measure for the dictionary and allows for a spatial kernel. This model is particularly suitable for hyperspectral images in which patterns vary slowly along the spatial dimensions.

- The thesis theoretically dismisses a developing concept that sparse representation may be inexpedient for classification [233]. It shows that accuracy of dense representation based classification can itself be improved by augmenting the dense representation with a sparse one.

- A generic Bayesian discriminative dictionary learning model is proposed that also results in automatic computation of dictionary size - the key parameter
in discriminative dictionary learning. Analytical expressions for performing Gibbs sampling [62] over the model are also derived.

- A Bayesian model for learning a classifier along a discriminative dictionary is proposed. The joint learning of the classifier customizes it to be used in conjunction with the dictionary for accurate classification.

- The joint model for learning discriminative dictionaries and classifiers is tailored to hyperspectral image classification by using Gaussian Processes [167] and contextual information in the sparse representation model.
PART I

Hyperspectral Unmixing
CHAPTER 2

Repeated Constrained Sparse Coding with Partial Dictionaries for Hyperspectral Unmixing

Abstract

Hyperspectral images obtained from remote sensing platforms have limited spatial resolution. Thus, each spectra measured at a pixel is usually a mixture of many pure spectral signatures (endmembers) corresponding to different materials on the ground. Hyperspectral unmixing aims at separating these mixed spectra into its constituent endmembers. We formulate hyperspectral unmixing as a constrained sparse coding (CSC) problem where unmixing is performed with the help of a library of pure spectral signatures under positivity and summation constraints. We propose two different methods that perform CSC repeatedly over the hyperspectral data. However, the first method, Repeated-CSC (RCSC), systematically neglects a few spectral bands of the data each time it performs the sparse coding. Whereas the second method, Repeated Spectral Derivative (RSD), takes the spectral derivative of the data before the sparse coding stage. The spectral derivative is taken such that it is not operated on a few selected bands. Experiments on simulated and real hyperspectral data and comparison with existing state of the art show that the proposed methods achieve significantly higher accuracy. Our results demonstrate the overall robustness of RCSC to noise and better performance of RSD at high signal to noise ratio.

Keywords : Hyperspectral unmixing, sparse coding, spectral derivative.

2.1 Introduction

Modern remote sensing instruments such as the NASA’s Airborne Visible Infrared Imaging Spectrometer (AVIRIS) [80] take hyperspectral (HS) images of Earth for geological studies. These images can be represented as HS cubes with two spatial and one spectral dimension (see Fig. 2.1). In typical remote sensing HS images, the spectral dimension consists of hundreds of contiguous bands in the visible and short infrared wavelength range. However, each pixel of the image cube represents a large area of the Earth’s surface in the spatial dimensions [20]. For instance, in the images captured by EnMAP HS imager of Germany [1] and Hyperion of NASA [145], each

\footnote{Published in Proc. of IEEE Winter Conf. on Applications of Computer Vision (WACV), 2014.}
2.1. Introduction

Figure 2.1: Illustration of hyperspectral image cube. The cube shows only eleven spectral bands (out of 224 bands) captured by AVIRIS. The image is taken over Cuprite mines, Nevada.

pixel represents about $30m^2$ area on the ground. Due to the low spatial resolution of HS sensors, the spectra measured by a sensor pixel is usually a mixture of spectra of different pure materials on the ground. Each spectra of pure material is called an endmember. Hyperspectral unmixing is the process of separating the measured spectra into its constituent endmembers and a set of fractional abundances of the corresponding materials (i.e. the proportion of each material in the image pixel), one set per pixel [25].

Hyperspectral unmixing is an active research area in remote sensing community [19], where it is seen as a blind source separation problem with the sources being statistically dependent [102]. Many works in hyperspectral unmixing exploit geometrical properties of the data in observed mixed pixels. These approaches are mainly based on the premise that in each mixed pixel the fractional abundances form a probability simplex and among a given collection of material spectra the constituent pixel endmembers can be found by estimating the smallest simplex set containing the observed pixel spectra [162], [160]. These methods assume presence of at least one pure pixel for every material captured in the image, thus requiring endmember extraction algorithms [102]. Vertex component analysis [146], N-FINDER [211], orthogonal subspace projection technique [168] and Pixel Purity Index [28] are some of the popular techniques and algorithms for endmember extraction for hyperspectral unmixing. The assumption of the existence of pure pixels in remote sensing HS cubes is not a practical one, therefore techniques like iterative error analysis [151], convex cone analysis [95], minimum volume simplex analysis [123] and minimum volume constrained non-negative matrix factorization [143] have been proposed to circumvent the problem by generating pure endmembers from hyperspectral images. However, these techniques are likely to fail in highly mixed scenarios, where the algorithms end up generating artificial endmembers that cannot be associated with spectral signatures of true materials [101].
To overcome the aforementioned issues, hyperspectral unmixing has recently been approached in a semi-supervised fashion [20], [19], [102], [97]. Under the assumption that an observed mixed pixel spectra can be represented as a linear combination of finite number of pure spectra of known materials, these approaches formulate hyperspectral unmixing as a sparse regression problem. They make use of spectral libraries of pure material made publicly available by US Geological Survey (USGS) [54] and Jet Propulsion Lab, NASA [14]. One major challenge faced by sparse regression approaches is the fact that the spectral libraries have very high mutual coherence between their elements [20], [19], [102]. Due to the large number of similar library elements (i.e. elements with high mutual coherence), a given mixed spectra can be represented with multiple different combinations of the library elements. Thus, detecting the correct linear combination of the endmembers, out of all possible linear combinations of the library elements, becomes a challenging problem. Researchers at German Aerospace Center (DLR) [20], [19] have recently proposed to increase the endmember detection rate by taking spectral derivatives of the library elements and the given HS image. Spectral derivative significantly reduces the mutual coherence of the library elements. This approach is able to increase the endmember detection rate but at the same time it is very sensitive to noise and works well only for HS images with very high Signal to Noise Ratio (SNR).

In this work we formulate hyperspectral unmixing as a constrained sparse coding problem and propose two different methods to improve the endmember detection rate. One of these methods shows better overall robustness to noise, whereas the other performs better at high SNR. In the first method, we propose to perform sparse coding of the hyperspectral cube repeatedly such that in each sparse coding step (except the first) few spectral bands in the library and the image are systematically neglected. Fractional abundances of the detected endmembers in each step are then combined in a weighted fashion. In the second method, sparse coding is again performed repeatedly. However, this time the coding is performed on the library and the image that is obtained by taking their spectral derivatives. The spectral derivatives are taken such that they are not operated on a few selected bands of the data. We perform experiments with the proposed methods on simulated data and real HS cube obtained by AVIRIS. We compare our results with the state of the art results shown by researchers at DLR in order to evaluate our methods. Experiments show improvements in the results with the proposed methods, especially for low SNR of hyperspectral images.

This paper is organized as following: after formulating the problem in Section 2.2,
we present the proposed methods in Section 2.3. Results of the experiments with the proposed methods and discussion on these results are given in Section 2.4. Section 2.5 concludes this work by summarizing the findings.

2.2 Problem Formulation

2.2.1 Linear Mixing Model

In this work, we focus on Linear Mixing Model (LMM) [25] for hyperspectral unmixing. This model assumes that at any given band in an HS cube, the spectral response of a pixel is a linear combination of all the constituent endmembers at that particular band. Written mathematically

\[ y_i = \sum_{j=1}^{p} l_{ij} \alpha_j + \epsilon_i, \]  

(2.1)

where \( y_i \) is the value of spectral reflectance measured at \( i^{th} \) spectral band, \( l_{ij} \) is the reflectance of the \( j^{th} \) endmember of the pixel at band \( i \), \( \alpha_j \) is the fractional abundance of the \( j^{th} \) endmember and \( \epsilon_i \) is the noise affecting the measurement. Assuming that HS image is acquired by a sensor with \( m \) spectral channels, LMM can be written in a matrix form:

\[ y = L \alpha + \epsilon, \]  

(2.2)

where \( y \in \mathbb{R}^{m \times 1} \) represents the measured reflectance at a pixel, \( L \in \mathbb{R}^{m \times p} \) is a matrix with \( p \) pure endmembers, \( \alpha \in \mathbb{R}^{p \times 1} \) is a vector with fractional abundances of the endmembers as its elements and \( \epsilon \in \mathbb{R}^{m \times 1} \) represents noise.

In a linear mixing model, fractional abundances of the constituent endmembers are subject to two constraints [25], (a) Abundance Non-negativity Constraint (ANC) \((\forall i, i \in \{1, ..., p\}, \alpha_i \geq 0)\) and (b) Abundance Sum-to-one Constraint (ASC) \((\sum_{i=1}^{p} \alpha_i = 1)\). These constraints owe to the fact that fractional abundances of the endmembers are non-negative quantities which, if detected exactly, sum up to 1 for the area represented by a pixel.

2.2.2 Hyperspectral Unmixing as Sparse Approximation

Let us denote a spectral library of materials by a matrix \( D \in \mathbb{R}^{m \times k} \) \((k > m)\) with each column \( d_i \in \mathbb{R}^{m \times 1} \) representing spectra of a pure material that is normalized to \( l_2 \) unit length. If we neglect noise, under LMM the spectral measurement \( y \) at any pixel of an HS image can be reconstructed with the spectral library:

\[ y = D \alpha. \]  

(2.3)
In practice, a given pixel ($30m^2$ ground area) contains only a limited number of materials. It would be safe to assume that a given sensed spectra will be a linear combination of no more than $p$ pure spectra, where $p \ll k$. Note that, in the literature related to sparse representation of images, $D$ is referred as dictionary and $d_i$ is termed as its atom. However, here we stick to the more common naming convention of remote sensing community and refer to $D$ as a library and $d_i$ as a pure spectral signature or an endmember.

With $k > m$, (2.3) represents an underdetermined system of equations that can have an infinite number of solutions. Therefore, instead of solving (2.3) we can minimize $||D\alpha - y||_2$, where $||.||_2$ is the $l_2$ Euclidean norm. Thus, we arrive at the following sparse approximation problem for hyperspectral unmixing:

$$\min ||\alpha||_0 \text{ s.t. } ||D\alpha - y||_2 \leq \eta,$$

(2.4)

where $\eta$ is some tolerance. Solving the above mentioned equation is NP-hard. However, its polynomial time approximation can be achieved by replacing $l_0$ minimizer with $l_1$ minimizer [48]. The sparse approximation problem can thus be re-written:

$$\min ||\alpha||_1 \text{ s.t. } ||D\alpha - y||_2 \leq \eta.$$

(2.5)

By including the Abundance Non-negativity Constraint in the above equation we arrive at the following constraint optimization problem:

$$\min ||\alpha||_1 \text{ s.t. } ||D\alpha - y||_2 \leq \eta \ \forall i, \alpha_i \geq 0.$$

(2.6)

Equation 2.6 can be solved using basis pursuit (BP) algorithm [48] or LASSO (least absolute shrinkage and selection operator) [194], with positivity constraint on the sparse vector coefficients. Previous works in hyperspectral unmixing (e.g. [19], [20]) formulate the problem as in Equation 2.6 and manually tune the value of $\eta$ to find the approximation which gives the best results. It is also possible to achieve the same results by solving the following problem instead of (2.6):

$$\min_\alpha ||D\alpha - y||_2 \text{ s.t. } ||\alpha||_1 \leq \lambda, \ \forall i, \alpha_i \geq 0.$$

(2.7)

The main advantage of formulating the hyperspectral unmixing problem as (2.7) is that we can use Abundance Sum-to-one Constraint to guess the value of $\lambda$ a priori.

2.2.3 Mutual Coherence

Mutual coherence of a library, $\mu(D)$, is defined as:

$$\mu(D) = \max_{i \neq j} |d_i^T d_j|.$$ 

(2.8)
2.2. Problem Formulation

![Image](image-url)

Figure 2.2: (a) Coherence matrix of the original library, with columns selected such that no two of them are more than 25 deg apart. (b) Coherence matrix of the differentiated library, with $c = 2$ in (2.9).

In sparse approximation techniques, small mutual coherence of the library is one of the most desirable conditions [19]. This is because, similar spectral signatures in the library can result in false detections of library elements in the sparse approximation process. However, in the case of large overcomplete libraries of material spectral signatures, high mutual coherence is unavoidable.

Iordache et. al. [102] show that for the value of ‘$p$’ to be as high as 10, considerable improvements can be achieved in sparse unmixing if the library is restricted to have spectral signatures that are different by 3 deg (i.e. $\mu(D) \leq 0.9986$). Bieniarz et al. [19] have proposed an approach that significantly reduces $\mu(D)$ by taking the spectral derivative of the library. Spectral derivative of a spectral signature $\mathbf{d} \in \mathbb{R}^{m \times 1}$, is defined as:

$$\Delta(\mathbf{d}) = \frac{d(b_i) - d(b_j)}{b_j - b_i}, \forall i, j \in \{1, ..., m - c\},$$

(2.9)

where, $b_k$ is the wavelength at $k^{th}$ band, $d(b_k)$ is the reflectance of the material at that wavelength and $j = i + c$, where $c = 1$. Fig. 2.2 shows an example of coherence reduction of a library using the spectral derivative. The image on the left shows the coherence matrix ($D^TD$) of a library with $\mu(D) = 0.9$. Coherence matrix of the differentiated library is represented by the image on the right.

It should be noticed that the operation of spectral derivative is very sensitive to noise. This is because, with a high spectral resolution of a hyperspectral sensor and $c = 1$ in (2.9), the value of spectral derivative at a band can change drastically with a small change in the reflectance value (due to noise) at that band or its adjacent band.
Chapter 2. Repeated Constrained Sparse Coding for Hyperspectral Unmixing

2.3 Proposed Solution

We propose two different algorithms for hyperspectral unmixing with constrained sparse coding. These algorithms are based on the following important observations. 1) For highly coherent spectral signatures, it is possible to reduce their mutual coherence by removing the bands from their spectra at which variance between the material reflectances is very low (see Fig. 2.3). 2) For spectral derivatives, adverse effects of noise on sparse unmixing can be mitigated by taking the spectral derivative such that it is not operated on the bands at which the material reflectances have very low variance across the spectral library. This happens because, at and near those bands, the highly coherent spectral signatures of materials are generally very similar to each other because of their inherent smooth nature [216]. Therefore, for noisy signals it is the noise that mainly contributes to the significant differences between the differentiated signals at those bands. Since, such differences can cause confusion in the unmixing process, we can simply neglect the aforementioned bands while taking the spectral derivative.

2.3.1 Repeated Constrained Sparse Coding (RCSC)

This method formulate hyperspectral unmixing as a constrained sparse coding problem and repeatedly solves the optimization function in Equation 2.7. The method is presented in Fig. 2.4 as an algorithm. RCSC first computes a sparse coefficient matrix $A_0$ with the help of the given library $D$ and the HS image $Y$. Without greatly affecting the spectral derivative’s ability of coherence reduction, we lower this sensitivity by using $c = 2$ in (2.9), which results in smoother spectra.
## 2.3. Proposed Solution

### 1. Algorithm RCSC: Inputs D, Y returns A

1. **Sparse code**: Compute $A_0$ with $D$ and $Y$, using (2.7).
2. **Cluster**: Cluster the columns of $D$ into $n$ clusters $C_i$,
   s.t. $\mu(C_i) \geq \cos \theta$, \forall $C_i$, $i \in \{1, \ldots, n\}$
3. for each $C_i$
   5. **Compute variance**: Compute variance of each row of $C_i$
   6. **Select rows**: Select ‘f’ fraction of rows with minimum variances.
   7. **Remove bands**: Create $\tilde{D}_i$ and $\tilde{Y}_i$ by removing rows corresponding to ‘f’ from $D$ & $Y$. 
   8. **Sparse code**: Compute $A_i$ with $\tilde{D}_i$ and $\tilde{Y}_i$ using (2.7).

### 9. $A = \frac{1}{2-\gamma} \sum_{i=0}^{n} \beta_i A_i$

10. return $A$

---

(Fig. 2.4, line ‘2’). Later on, it computes ‘n’ different sparse coefficient matrices $A_i$, where $i \in \{1, \ldots, n\}$, such that an $A_i$ is computed with a library $\tilde{D}_i$ and an image $\tilde{Y}_i$ that are obtained by neglecting ‘f’ fraction of bands from $D$ and $Y$, respectively (illustrated in Fig. 2.5). The algorithm uses all the intermediate sparse coefficient matrices to compute the final sparse coefficient matrix $A$, whose columns represent the fractional abundances of the detected materials in the pixels of the HS image. Given a library $D$ and the image $Y$, the bands that are neglected in computing an $A_i$ are selected as follows. The algorithm first clusters the spectral signatures in $D$ (Fig. 2.4, line ‘3’). Each cluster $C_i$ ($i \in \{1, \ldots, n\}$) is created as an $m \times c$ matrix such that, $c > 2$ and $\mu(C_i) \geq \cos \theta$, where $\theta$ denotes the threshold angle (i.e. the maximum angle allowed between any two spectral signatures in $C_i$). For each $C_i$, the algorithm selects ‘f’ fraction of its rows with minimum variances. The rows corresponding to ‘f’ are removed from $D$ and $Y$ to obtain $\tilde{D}_i$ and $\tilde{Y}_i$ (Fig. 2.4, line ‘7’). Since the rows of these matrices correspond to spectral bands, removing them simply means neglecting the corresponding spectral bands in the HS data. The algorithm computes a sparse coefficient matrix $A_i$ with the corresponding $\tilde{D}_i$ and $\tilde{Y}_i$ for each $C_i$ (Fig. 2.4, line ‘8’). It should be noticed that we use $D$ and $Y$ to compute $A_0$, but reduced matrices $\tilde{D}_i$s and $\tilde{Y}_i$s to compute all other $A_i$s. This implies that all the learned sparse coefficient matrices may be different, however they have the same dimensions. After exhausting the list of the clusters, RCSC directly combines all the sparse coefficient matrices in a weighted fashion for calculating the final fractional abundance matrix $A$ (Fig. 2.4, line ‘9’), where:

\[
\beta_i = \begin{cases} 
1 & \text{if } i = 0 \\
(1 - f)/n & \text{if } i \neq 0
\end{cases}
\]
Chapter 2. Repeated Constrained Sparse Coding for Hyperspectral Unmixing

Figure 2.5: Illustration of computing ‘n’ sparse coefficient matrices with RCSC with a single library. The library is first clustered into ‘n’ clusters. Each cluster is then used to create a library $\tilde{D}_i$ by removing the bands which have least variance in the cluster. With each library, a different sparse coefficient matrix $A_i$ is computed.

There are two main advantages of solving (2.7) in a repeated manner as described above. 1) Each time when we drop spectral bands of the library, mutual coherence of highly similar spectral signatures belonging to the cluster under consideration, reduces. This improves $A_i$ if any of the constituent endmembers of the mixed pixels belongs to the cluster under consideration. 2) Even if none of the constituent endmembers belong to the current cluster, each time the sparse coding step converges to a slightly different sparse coefficient matrix. With our settings, each one of these matrices by itself achieves a reasonable endmember detection rate\(^1\). Therefore, combining all the computed sparse coefficient matrices results in even better endmember detection.

2.3.2 Repeated Spectral Derivative (RSD)

This method uses the concept of spectral derivative (see section 2.2.3) and solves Equation 2.7 with an image $\hat{Y}_i$ and a library $\hat{D}_i$ to obtain a sparse coefficient matrix $A_i$. This is repeated ‘n’ times, such that each time $\hat{Y}_i$ and $\hat{D}_i$ represent different matrices which are obtained by taking spectral derivatives of $Y$ and $D$. Each time the spectral derivative is taken such that it is operated on all the spectral bands in $Y$ and $D$ except at a small fraction ‘f’ of them. As in RCSC, the value of ‘n’ and the bands that correspond to ‘f’ are found by clustering $D$ into ‘n’ clusters stored in matrices $C_i$ ($i \in \{1, \ldots, n\}$). Once again, the bands that belong to ‘f’ are those which correspond to the rows of $C_i$ that have least variances. In $\hat{Y}_i$ and $\hat{D}_i$, the rows representing these bands are kept the same as those in $Y$ and $D$. Since ‘f’ represents a small fraction, it is also possible to simply drop these bands from the

\(^1\)We make this observation based on the results of our further experiments not reported here.
2.4. Results and Discussion

This section presents the results of applying the proposed methods to simulated data as well as real data acquired by the AVIRIS. In order to evaluate our methods we compare the results with the state of the art approach proposed in [19].

2.4.1 Spectral Library

We use the library of spectral signatures made publicly available by the NASA’s Jet Propulsion Lab (http://speclib.jpl.nasa.gov/). This library, known as the ASTER spectral library [14], consists of spectra of 2400 materials of seven different types (e.g. minerals, rocks). In our experiments we take a subset of this library that consists of 500 materials which belong to type mineral, rock, soil and vegetation. This subset, D (henceforth referred as the ‘library’) was created such that no two spectral signatures in it are more than 25 deg apart. Such a high mutual coherence between all the elements of the library is a rather strict condition. However, we impose this condition for better evaluation of the proposed methods, which aim
Chapter 2. Repeated Constrained Sparse Coding for Hyperspectral Unmixing

2.4.2 Simulated Data

We first test the proposed methods on simulated data, for which we create synthetic image cubes of dimensions $100 \times 100 \times 224$, where 224 is the spectral dimension. Each pixel of a synthetic HS cube is created by mixing $p$ randomly selected signatures from the library. Values of fractional abundances associated with each spectra are also selected randomly such that the ASC holds for each pixel. After creation of a cube, we include additive white Gaussian noise. Each result presented below is the mean value calculated with ten synthetic HS cubes. In our experiments, we use $\lambda = 1.3$ in Equation 2.7. However, the results are relatively insensitive to the value of $\lambda$ in the range $[1.1, 1.5]$. This range is based on ASC and allowing for any possible false detection of endmembers because of noise and high mutual coherence of the library. We select $\lambda = 1.3$ with three fold cross-validation. Value of $f$ in RCSC and RSD is kept at 0.1, whereas $\theta$ is chosen to be 7 deg.

Fig. 2.7 shows results of applying RCSC and RSD to the simulated data. The figure also includes results of applying the approach in [19] (shown as DLR in the figure) and directly performing CSC on the same data. For DLR, we use our own implementation which was done with the help of the authors of [19]. Fig. 2.7a shows the mean values of endmember detection rate as a function of SNR. Here, endmember detection rate is defined as the percentage of endmembers in a mixed pixel that have been correctly detected by an algorithm. The graph shows better endmember detection rates, especially at low SNR, for the proposed methods. When a sparse approximation method is used for hyperspectral unmixing, it can also result in false detections of endmembers. In order to properly evaluate a method, it is important to note the fractional abundances of the materials which have been falsely detected by the method. Fig. 2.7b shows the comparison of fractional abundances in the false detections (as percentage) for the methods, as a function of SNR. Here, curves for RCSC and RSD are similar to those of CSC and DLR respectively, which implies that both of the proposed methods are able to improve the endmember detection rate without significantly incurring false detections in the repetitions. In fact, for low SNR RSD outperforms DLR. This is because, while taking the spectral derivative RSD neglects the bands in the data that cause confusion in unmixing (see Fig. 2.6, line ‘6’). For RSD, fractional abundance in the false detections become
2.4. Results and Discussion

Figure 2.7: Comparison of the results of RCSC, RSD, CSC and DLR [19]. (a) Comparison of the mean endmember detection rate as a function of SNR (with \( p = 5 \)). (b) Comparison of mean fractional abundances in false detections of endmembers with increasing SNR (with \( p = 5 \)). (c) Comparison of the mean endmember detection rate as a function of cardinality of image pixels at 50 dB SNR.

almost zero beyond 110 dB, whereas they remain of the order of 3 for RCSC.

For a typical HS unmixing scenario in real images, the cardinality ‘\( p \)’ of a mixed pixel is generally of the order of five [102]. Therefore, the above mentioned results are evaluated with \( p = 5 \) for each pixel of each image. For different areas on land, the value of \( p \) can vary for different pixels. Fig. 2.7c shows the endmember detection rate for each method as a function of cardinality of the pixels of the images. Here, SNR is fixed at 50 dB. In the figure, the proposed methods clearly outperform CSC and DLR for all of the values of \( p \). Notice that, the results for CSC, RCSC and RSD shown in Fig. 2.7 are obtained under our settings. That is, for each of these methods we solve for Equation 2.7 with \( \lambda = 1.3 \), while performing sparse coding. On the other hand, the results for DLR are obtained by solving for Equation 2.6, with \( \eta = 10^{-6} \) as in [19]. It is interesting to know the effects of formulating the sparse coding

\(^2\)Under our settings results of DLR are worse than those reported here.
Chapter 2. Repeated Constrained Sparse Coding for Hyperspectral Unmixing

Figure 2.8: Comparison of optimization Equation 2.6, with $\eta = 10^{-6}$ and Equation 2.7, with $\lambda = 1.3$. (a) Compares the mean endmember detection rate as a function of SNR (with $p = 5$). (b) Shows mean fractional abundances in false detections of endmembers with increasing SNR (with $p = 5$).

problem as Equation 2.7 instead of Equation 2.6. For both of these equations, Fig. 2.8 shows the endmember detection rate (Fig. 2.8a) and the fractional abundance in false detections (Fig. 2.8b) as a function of SNR. In the figure, there is a clear separation between the curves at high SNR. The main reason behind this separation is that Equation 2.6 minimizes the $l_1$ norm of the coefficient vector without any explicit bounds on it. This limits its performance at high SNR where high mutual coherence between the pure spectral signatures result in false detection of similar endmembers within the tolerance allowed for the reconstruction error. On the other hand, Equation 2.7 explicitly constraints the $l_1$ norm of the coefficient vector through $\lambda$, which is justified by ASC. Thus, minimizing the reconstruction errors without explicit lower bounds result in a better performance at high SNR.

2.4.3 Real Data

We apply RCSC and RSD on the real HS image collected by AVIRIS (http://aviris.jpl.nasa.gov/data/free$_$data.html). From this image, we selected an HS cube of dimension $614 \times 512 \times 224$. The spatial dimensions ($614 \times 512$) of this cube represent a region of Cuprite mines, Nevada which has been well studied in Geological Sciences literature for its surface materials. Fig. 2.9a shows material classification results of this region from [54] which are generally used as a benchmark for qualitative evaluation of hyperspectral unmixing approaches in remote sensing community. To apply our methods to real data, we first drop 24 spectral bands in the HS cube (and the library) that have zero or very low reflectance values due to atmospheric absorptions. Moreover, we rely on advanced atmospheric correction
2.4. Results and Discussion

Figure 2.9: Results on real data: (a) Classification results of Cuprite mines, Nevada from [54]. To evaluate the proposed approaches we use the HS cube taken over the region magnified in the image. (b) Abundance maps for K-Alunite created by RSD (left) and RCSC (right) for the region magnified in Fig. 2.9a. 

Algorithms to convert the at-sensor radiance measurement by AVIRIS to reflectance units in order to match spectral signatures in the library, which are measured in laboratory conditions. These algorithms have been applied to the available image.

Fig. 2.9b shows the abundance maps created by RSD (left) and RCSC (right) of a material ‘K-Alunite’ present in the region. The region shown in the figure corresponds to the region magnified in Fig. 2.9a. As can be seen in the abundance maps, most of the ‘K-Alunite’ has been correctly identified by both of the methods. However, results of RSD are better in the sense that the algorithm does not over estimate the presence of ‘K-Alunite’. This is the consequence of coherence reduction with spectral derivative. RCSC incorrectly detects ‘K-Alunite’ at some regions where the materials with similar spectral signatures are present. For instance, it also detects ‘K-Alunite’ in some regions which have been classified as ‘Alunite+Kaolinite and/or Muscovite’ and ‘Kaolinite’ according to [54].
2.5 Conclusion

In this work, we formulate hyperspectral unmixing as a constrained sparse coding problem and propose two different methods to achieve high endmember detection rates, especially with low SNR of HS images. Both of the methods solve for constrained sparse coding optimization function in a repeated manner. In the first method, called RCSC, we systematically remove a small fraction of spectral bands from the data each time before computing a sparse coefficient matrix. The final fractional abundance matrix is then computed as a weighted sum of all the computed sparse coefficient matrices. In the second method, called RSD, we make use of spectral derivative, which is operated on all the bands of the data except a small fraction of them. After sparse coding with the differentiated data each time, we calculate the fractional abundance matrix as the mean of the sparse coefficient matrices obtained in each sparse coding step. We apply the methods to both simulated as well as real HS images. We compare the results of hyperspectral unmixing with a state of the art approach proposed in [19]. Results show better performance of the proposed methods, especially at low SNR of HS images.
CHAPTER 3

Futuristic Greedy Approach to Sparse Unmixing of Hyperspectral Data

Abstract

Spectra measured at a single pixel of a remotely sensed hyperspectral image is usually a mixture of multiple spectral signatures (endmembers) corresponding to different materials on the ground. Sparse unmixing assumes that a mixed pixel is a sparse linear combination of different spectra already available in a spectral library. It uses sparse approximation techniques to solve the hyperspectral unmixing problem. Among these techniques, greedy algorithms suite well to sparse unmixing. However, their accuracy is immensely compromised by the high correlation of the spectra of different materials. This work proposes a novel greedy algorithm, called OMP-Star, that shows robustness against the high correlation of spectral signatures. We preprocess the signals with spectral derivatives before they are used by the algorithm. To approximate the mixed pixel spectra, the algorithm employs a futuristic greedy approach that, if necessary, considers its future iterations before identifying an endmember. We also extend OMP-Star to exploit the non-negativity of spectral mixing. Experiments on simulated and real hyperspectral data show that the proposed algorithms outperform the state of the art greedy algorithms. Moreover, the proposed approach achieves results comparable to convex relaxation based sparse approximation techniques, while maintaining the advantages of greedy approaches.

Keywords: Hyperspectral unmixing, sparse unmixing, greedy algorithm, orthogonal matching pursuit.

3.1 Introduction

Hyperspectral remote sensing extracts information from the scenes on the Earth’s surface, using the radiance measured by airborne or spaceborne sensors [24], [35]. These sensors measure the spectra of the Earth’s surface at hundreds of contiguous narrow bands [189], resulting in a hyperspectral data cube that has two spatial and one spectral dimension (see Fig. 3.1). Each pixel of such a data cube is a vector that represents the spectral signature of the objects/materials measured at

Chapter 3. Futuristic Greedy Approach to Sparse Unmixing

Figure 3.1: Illustration of hyperspectral data cube: The XY-plane corresponds to spatial dimensions. The spectral dimension shows the data collected at different wavelengths. The cube is illustrated with only eleven spectral bands (out of 224). The data is taken over Cuprite mines, Nevada by AVIRIS [80].

The pixel. Due to low spatial resolution of sensors, presence of intimate mixtures of materials and multiple scattering, the signature of the pixel is usually a combination of several pure spectral signatures. Each of these pure spectral signatures is called an endmember. Hyperspectral unmixing aims at extracting these endmembers and their fractional abundances (i.e. proportion of the endmembers in a pixel), one set per pixel [25].

In recent years, linear unmixing of hyperspectral data has attracted significant interest of researchers [24]. Linear unmixing assumes that a mixed spectra in a hyperspectral data cube can be expressed as a linear combination of the endmembers, weighted by their fractional abundances. Many works under this model have exploited the geometric properties of the hyperspectral data (e.g., [27] - [57]). Such approaches exploit the fact that the convex hull of pure endmembers in the data forms a probability simplex. Thus, finding the endmembers amounts to finding the vertices of the simplex [160], [162]. Most of the classical geometrical methods for unmixing assume the presence of at least one pure pixel for every material captured in the scene. Vertex Component Analysis [146], Pixel Purity Index [28], Simplex Growing Algorithm [43], Successive Volume Maximization [42], N-FINDR [211], Iterative Error Analysis [151] and Recursive Algorithm for Separable Non-negative Matrix Factorization [78] are popular examples of such methods.

The assumption of the existence of pure spectra in a hyperspectral data cube is not a practical one. Therefore, techniques like Minimum Volume Simplex Analysis [123], Minimum Volume Transform-Nonnegative Matrix Factorization [143], Iterative Constrained Endmembers (ICE) [17] and Sparsity Promoting ICE [230] have been proposed to circumvent the problem by generating pure endmembers from the hyperspectral data itself. However, these techniques are likely to fail in highly mixed
scenarios, where the algorithms end up generating artificial endmembers that cannot be associated with the spectral signatures of true materials [101]. For such cases, hyperspectral unmixing is usually formulated as a statistical inferencing problem and Bayesian paradigm becomes the common choice [24]. Under this paradigm, computational complexity of Bayesian inference becomes a bottleneck for effective hyperspectral unmixing.

In order to overcome the aforementioned issues, hyperspectral unmixing has recently been approached in a semi-supervised fashion [102]. This approach formulates hyperspectral unmixing as a Sparse Approximation (SA) problem and aims at developing efficient and accurate SA algorithms for sparse unmixing. Sparse unmixing makes use of a library of pure spectra and finds the optimal subset from the library that can best model the mixed pixel [25]. Iordache et al. [102], have studied different SA algorithms for sparse unmixing of hyperspectral data, including Orthogonal Matching Pursuit (OMP) [33], Basis Pursuit (BP) [48], BP denoising (BPDN) [48] and Iterative Spectral Mixture Analysis (ISMA) [173]. Recently, the authors have also exploited the spatial information [98] and the subspace nature [99] of the hyperspectral data in sparse unmixing. Previous works in sparse unmixing have mainly focused on SA algorithms that are based on convex relaxation of the problem. Generally, the sparse approximation algorithms based on the greedy approach [33] have lower computational complexity than their convex relaxation counterparts [31], [183]. These algorithms find approximate solution for the \( l_0 \) problem directly, without smoothing the penalty function [183]. Furthermore, the greedy algorithms admit to simpler and faster implementation [196]. However, Iordache et al. [102] has shown that in comparison to the convex relaxation algorithms the accuracy of the greedy algorithms (e.g., OMP) is adversely affected by the high correlation between the spectra of the pure materials.

Shi et al. [183] have strongly argued to exploit the potential of the greedy approach for sparse unmixing and propose a greedy algorithm, called Simultaneous Matching Pursuit (SMP). This algorithm processes the data in terms of spatial blocks and exploits the contextual information in the data to mitigate the problems caused by the high correlation among the spectra. However, the block size becomes an important parameter for SMP that is specific to hyperspectral data cube. Furthermore, the work assumes the existence of only a few endmembers in the whole data cube and the presence of spatial information everywhere in the data. Since hyperspectral unmixing is primarily a pixel-based process, we favor pixel-based greedy algorithms for sparse unmixing. These algorithms do not assume the existence of
contextual information in the data. However, they can always be enhanced to take advantage of the contextual information following the guidelines in [199].

In this work, we propose a greedy algorithm for sparse unmixing, called OMP-Star. OMP-Star is a pixel-based algorithm that augments OMP’s greedy pursuit strategy with a futuristic heuristic. This heuristic is inspired by a popular search algorithm, called A-Star [85]. OMP-Star shows robustness against the problems caused by the high correlation among the spectra while maintaining the advantages of greedy algorithms. We further modify the proposed algorithm such that it takes advantage of the non-negative nature of the fractional abundances. This constrained version of the proposed algorithm is called OMP-Star+. The second contribution of this work is that it exploits derivatives [200] of hyperspectral data for sparse unmixing with greedy algorithms. It is possible to reduce correlation among spectral signatures by taking their derivatives [4], [19]. Therefore, we preprocess hyperspectral data with derivatives for all the greedy algorithms. Although preprocessing the data reduces the correlation among the spectra, yet the reduction is generally not sufficient to achieve accurate results using the greedy algorithms. Therefore, a greedy algorithm with robustness against high signal correlation remains desirable for hyperspectral unmixing. This article also makes a minor theoretical contribution to sparse unmixing by deriving the condition for spectral libraries which ensures that the non-negativity of fractional abundances automatically constrains their sum to constant values. This is important because once a spectral library is processed to satisfy this condition, we do not need to explicitly impose the aforementioned constraint, which must be satisfied by the computed fractional abundances. We test the proposed approach thoroughly on simulated as well as real hyperspectral data. The results show that the proposed algorithms show better performance than the existing greedy algorithms. Furthermore, the proposed approach achieves results comparable to the convex relaxation based sparse approximation algorithms, with a considerable computational advantage.

This article is organized as follows: Section 3.2 formulates hyperspectral unmixing as a sparse approximation problem and presents the aforementioned theoretical contribution of this work. In Section 3.3, we review the important greedy algorithms for sparse approximation. Section 3.4 presents the proposed algorithms. We study derivatives for sparse unmixing in Section 3.5. The proposed algorithms are evaluated with synthetic data in Section 3.6 and with real data in Section 3.7. The computational complexity analysis of the proposed algorithms is presented in Section 3.8. We draw conclusions in Section 11.6.
3.2 Hyperspectral Unmixing as Sparse Approximation

3.2.1 Linear Mixing Model

Hyperspectral unmixing as a sparse approximation problem focuses on Linear Mixing Model (LMM). LMM assumes that the spectral response at a band in the mixed pixel is a linear combination of the constituent endmembers at that band. Written mathematically,

\[ y_i = \sum_{j=1}^{p} l_{ij} \alpha_j + \epsilon_i, \]  

where \( y_i \) is the spectral reflectance measured at the \( i^{th} \) band, \( p \) is the total number of endmembers in the pixel, \( l_{ij} \) is the reflectance of the \( j^{th} \) endmember at band \( i \), \( \alpha_j \) is the fractional abundance of the corresponding endmember and \( \epsilon_i \) is the noise affecting the measurement. Assuming that the hyperspectral data cube is acquired by a sensor with \( m \) spectral channels, LMM can be written in a matrix form:

\[ y = L \alpha + \epsilon, \]  

where \( y \in \mathbb{R}^m \) represents the measured reflectance at a pixel, \( L \in \mathbb{R}^{m \times p} \) is a matrix containing the endmembers, \( \alpha \in \mathbb{R}^p \) is a vector with fractional abundances of the corresponding endmembers and \( \epsilon \in \mathbb{R}^m \) represents noise.

In LMM, fractional abundances of the endmembers must satisfy two constraints \cite{24} (a) ANC: Abundance Non-negativity Constraint (\( \forall i, i \in \{1, \ldots, p\}, \alpha_i > 0 \)) and (b) ASC: Abundance Sum-to-one Constraint (\( \sum_{i=1}^{p} \alpha_i = 1 \)). These constraints owe to the fact that the fractional abundances of the endmembers are non-negative quantities that sum up to 1 for each pixel.

3.2.2 Sparse Unmixing

Let us denote a spectral library by a matrix \( D \in \mathbb{R}^{m \times k} \) (\( k > m \)) with each column \( d_i \in \mathbb{R}^m \) representing the spectral signature of a pure material. Under the assumption that \( D \) includes all the endmembers of a pixel, the signal \( y \) at the pixel can be reconstructed as:

\[ y = D \alpha + \epsilon, \]  

where \( \alpha \in \mathbb{R}^k \) has only \( p \) non-zero elements (\( p \ll k \)). Without considering the above mentioned constraints over the fractional abundances, sparse unmixing is formulated as the following energy optimization problem:

\[ (P_0^n): \min_{\alpha} \|\alpha\|_0 \quad s.t. \quad \|D \alpha - y\|_2 \leq \eta, \]
where $||.||_0$ is the $l_0$ pseudo-norm that simply counts the number of non-zero elements in $\alpha$, and $\eta$ is the tolerance due to noise and modeling error.

$(P_0^\eta)$ is generally an NP-hard problem [148] and in practice, it is usually solved with the greedy sparse approximation algorithms (e.g., OMP) or by convexification of the problem (e.g., BP). In sparse unmixing, the support of the solution $\alpha$ denotes the indices of the endmembers in the spectral library $D$. Note that, the goal of hyperspectral unmixing is to find (a) the correct set of endmembers and (b) their corresponding fractional abundances. This is different from the goal of high fidelity sparse approximation of $y$ using $D$. In fact, with similar spectral signatures in $D$, a good solution to the later may be practically useless for the former.

Introducing the Abundance Non-negativity Constraint in $(P_0^\eta)$ gives us the following formulation:

$$(P_0^{\eta+}) : \min_{\alpha} ||\alpha||_0 \ s.t. \ ||D\alpha - y||_2 \leq \eta, \ \alpha > 0,$$

($(P_0^{\eta+})$ is constrained by ANC and not by ASC. Previous works in sparse unmixing (e.g., [102] and [183]) state that imposing ANC in sparse unmixing automatically imposes a general version of ASC on the problem. This may not always be the case, however if $D$ is normalized in $l_1$-norm then the statement is true in general\footnote{Here, we do not claim incorrectness of the previous works, but we emphasize the importance of normalization. Using $D$ without normalizing the spectra, may not automatically guarantee the imposition of the generalized ASC along with ANC. This fact has not been stated clearly in the previous literature related to sparse unmixing.}. We augment this argument with analytical results presented in Section 3.2.3. The general version of ASC implies, for a sparse unmixing solution $\alpha$, $||\alpha||_1 = c$ [102]. Here, $||.||_1$ represents the $l_1$-norm of the vector and $c$ is a pixel-dependent scale factor. The generalized ASC simply becomes ASC when $c = 1$.

Minimization of $l_0$-pseudo norm of $\alpha$ is generally performed with greedy sparse approximation algorithms. Relaxed convexification of this minimization problem has also been widely researched. With relaxed convexification, the sparse unmixing problem can be re-written as:

$$(P_1^{\eta+}) : \min_{\alpha} ||\alpha||_1 \ s.t. \ ||D\alpha - y||_2 \leq \eta, \ \alpha > 0.$$  

($(P_1^{\eta+})$ is more tractable than $(P_0^{\eta+})$ because of the convex nature of $l_1$-norm [70]. In the context of sparse unmixing, Constraint Spectral Unmixing by variable Splitting and Augmented Langrangian (CSUnSAL) [22] is a popular algorithm for solving $(P_1^{\eta+})$. CSUnSAL exploits the Alternating Direction Method of Multipliers (ADMM) presented in [68] and it is tailored for hyperspectral unmixing.
If we neglect ANC in \((P_1^{n+})\) then the rest of the problem is equivalent to the well known LASSO [194] problem, given below as \((P_1^\lambda)\). Of course, we need an appropriate Langrangian multiplier \(\lambda\) for this equivalency.

\[
(P_1^\lambda): \min_{\alpha} \frac{1}{2} ||y - D\alpha||_2 + \lambda ||\alpha||_1. \tag{3.7}
\]

Iordache et al. [102] used Sparse Unmixing by variable Splitting and Augmented Langrangian (SUnSAL) [22] to solve \((P_1^\lambda)\) for hyperspectral unmixing. Similar to CSUnSAL, SUnSAL exploits ADMM for sparse unmixing. Recent works [98], [235] in sparse unmixing, focussing on convex relaxation of the problem, have also exploited the piece-wise smoothness of fractional abundances in hyperspectral images. More specifically, they add a total variation regularizer in \((P_1^{n+})\). Similarly, Iordache et al. [99] have also made use of the subspace nature of hyperspectral image for improved results via SUnSAL. In [99], SUnSAL is extended to collaborative SUnSAL (CLSUnSAL) that solves \((P_1^\lambda)\) under structured sparse approximation framework.

### 3.2.3 Automatic Imposition of ASC with ANC

**Theorem (3.1)** For a given spectral library \(D \in \mathbb{R}^{n \times k}\), if \(\exists h > 0\), such that \(h^T D = l 1^T\) for \(l > 0\), then \(\alpha \in \{y = D\alpha, \alpha > 0\}\) also satisfies \(||\alpha||_1 = c\), where \(h \in \mathbb{R}^{m \times 1}, 1 \in \mathbb{R}^{k \times 1}\) is a vector of 1s and \(c\) is a constant.

**Proof:** Consider the following model that admits to a sparse solution:

\[
D\alpha = y, \quad \alpha > 0. \tag{3.8}
\]

Given that \(D\) is a non-negative matrix, \(\exists h \in \mathbb{R}^{m \times 1}\) such that:

\[
h^T D\alpha = h^T y = v, h > 0, v > 0. \tag{3.9}
\]

Imposing that a valid solution of (3.8) also satisfies \(||\alpha||_1 = c\), where \(c > 0\), then:

\[
l 1^T \alpha = v, l > 0, \tag{3.10}
\]

where, \(1 \in \mathbb{R}^{k \times 1}\) is a vector of 1s and \(l = v/c\). From (3.9) and (3.10) we arrive at:

\[
h^T D = l 1^T, l > 0. \tag{3.11}
\]

We can reach (3.11) only under the assumption that \(||\alpha||_1 = c\). Hence, for a given \(D\), if \(\exists h\) such that (3.11) is satisfied, a solution to (3.8) will also satisfy \(||\alpha||_1 = c\).

**Corollary(1):** If \(d_i\) represents the \(i^{th}\) column of \(D\) and

\[
||d_i||_1 = l \forall i \in \{1, ..., k\}, l > 0, \tag{3.12}
\]
then, (3.11) is always satisfied by $h = 1$, where $1 \in \mathbb{R}^{m \times 1}$ is a vector of ones.

**Corollary (2):** a) If the columns of $D$ are normalized in $l_1$-norm then $||\alpha||_1 = c = ||y||_1$ - automatic imposition of the generalized ASC due to ANC. Here, $||y||_1$ is a pixel dependent scaling factor. 

b) If the columns of $D$ are normalized in $l_1$-norm and scaled by $||y||_1$ then $||\alpha||_1 = 1$ - automatic imposition of ASC due to ANC.

The result in Corollary 2a follows from the reasoning: When $D$ is normalized in $l_1$-norm, $l = 1$ in (3.12). Therefore, (3.11) is always satisfied by $h = 1$ and $1^Ty = ||y||_1 = v$ according to (3.9). This makes $c = v$ (since $l = v/c$) and $1^T\alpha = ||\alpha||_1 = c$ according to (3.10). Similarly, with $l = ||y||_1$ in (3.12), we finally get $||\alpha||_1 = c = 1$. Note that, for hyperspectral unmixing problem, generally $\not\exists h$ that satisfies (3.11). However, using the above results we can always process $D$ to ensure that generalized ASC is automatically satisfied by a solution of (3.8). This processing requires simple $l_1$-normalization of the library according to Corollary 2.

Previous works in sparse unmixing have mentioned Elad’s results in [31] for automatic imposition of generalized ASC due to ANC. The above conclusions are inline with those results. Appendix-A shows how we can extend Elad’s results to arrive at the conclusion in Corollary 2a. However, the analysis presented in this section is more general and the results shown here subsume those in Appendix-A.

### 3.3 Greedy Algorithms

Greedy algorithms provide a polynomial time approximation of $(P_\eta^0)$ by iteratively selecting the columns of $D$ that best approximate $y$. We first discuss Orthogonal Matching Pursuit (OMP) [33] as the representative algorithm, which is given in Algorithm 1. Each iteration of OMP can be divided into three steps. In the first step, OMP identifies the column in $D$ that minimizes the residue of the current approximation of $y$ (identification), where $y$ itself is considered as the residual vector in the first iteration. The identified column is added to a selected subspace (augmentation). Next, a new residual vector is computed by approximating $y$ in the column span of the selected subspace and subtracting this approximation from $y$ (residual update). The above three steps are repeated until the stopping rule is satisfied. OMP updates the residual vector by computing the least squares approximation of $y$ with the selected subspace, which makes the updated residual vector orthogonal to the selected subspace. Therefore, each newly identified column of $D$ is different from those already present in the selected subspace. OMP’s procedure of residual update is an enhancement over Matching Pursuit (MP) algorithm [138] and also the reason why it is called the Orthogonal MP. The MP algorithm updates
Algorithm 1 OMP

**Initialization:**
1: Iteration: $i = 0$
2: Initial solution: $\alpha^0 = 0$
3: Initial residual: $r^0 = y - D\alpha^0 = y$
4: Selected support: $\mathcal{S}^0 = \text{support}\{\alpha^0\} = \emptyset$

**Main Iteration:** Update iteration: $i = i + 1$

**Identification:**
5: Compute $\epsilon(j) = \min_{z_j} \|d_j z_j - r^{i-1}\|_2^2$, $\forall j \in \{1, ..., k\}$, using the optimal choice $z_j^* = \frac{d_j r^{i-1}}{\|d_j\|_2^2}$
6: Find a minimizer, $j_0$ of $\epsilon(j) : \forall j \notin \mathcal{S}^{i-1}, \epsilon(j_0) \leq \epsilon(j)$

**Augmentation:**
7: $\mathcal{S}^i = \mathcal{S}^{i-1} \cup \{j_0\}$

**Residual update:**
8: Compute $\alpha^i = \min_{\alpha} \|D\alpha - y\|_2^2$ s.t. $\text{support}\{\alpha^i\} = \mathcal{S}^i$
9: $r^i = y - D\alpha^i$

**Stopping rule:**
10: If $\|r^i\|_2 < \epsilon_0$, stop. Otherwise iterate again.

the residual vector by simply deflating it with the recently identified column of $D$.

Different enhancements over OMP have also been proposed in the literature related to image and signal processing. A non-negative variant of OMP, henceforth denoted as OMP+, is proposed in [31]. OMP+ differs from OMP mainly in the residual update step, where the vector $\alpha^i$ is constrained to have only positive elements in line ‘8’ of Algorithm 1. Wang et al. [205] also proposed a generalized version of OMP, called generalized-OMP (gOMP). Difference between OMP and gOMP is in the identification and augmentation steps. Instead of identifying a single column of $D$, gOMP identifies $L$ columns in each iteration and augments the selected subspace with all of these vectors. Here, $L$ is an algorithm parameter. Lookahead-OMP (LAOMP) [45] is a variant of OMP that modifies the identification step. LAOMP also identifies $L$ (a prefixed number) columns in the identification step. Then, it picks one of these vectors and temporarily augments the selected subspace with it. It then proceeds similar to OMP until the stopping rule is satisfied. At the end, it stores the left-over residue. This procedure is repeated for all the $L$ identified columns. LAOMP then permanently augments the selected subspace with the col-
umn that resulted in the least left-over residue, neglecting the other \( L - 1 \) vectors. A*OMP [108] is another variant of OMP that directly integrates the A-Star search [85] in OMP. We defer further discussion on this approach to Section 3.4, where we compare and contrast this approach with the proposed algorithm.

Subspace Pursuit (SP) [60], Compressive Sampling MP (CoSaMP) [150] and Regularized-OMP (ROMP) [149] are the greedy algorithms that assume prior knowledge of the cardinality \( p \) of \( y \). All of these algorithms identify multiple columns of \( D \) in the identification step. In each iteration, SP identifies \( p \) columns and augments the selected subspace with all of them. It then approximates \( y \) with the augmented subspace in the least squares sense. Then, it selects the vectors of the augmented subspace that correspond to the coefficients of the solution with \( p \) largest magnitudes. These vectors compose the updated selected subspace. Once the selected subspace is updated, SP updates the residual vector like OMP. CoSaMP identifies \( 2p \) columns in each iteration and augments the selected subspace with all of them. Then it selects \( p \) vectors like SP. However, it updates the residual vector by using the already computed coefficients of the selected \( p \) vectors. ROMP also identifies \( p \) columns in each iteration. Then it drops off some of these vectors using a predefined regularization rule before the augmentation step. In ROMP, the residual vector is also updated following OMP. The algorithms mentioned in this paragraph converge to solutions very quickly. However, the assumption of prior knowledge of the mixed signal’s cardinality compromises their practicality for sparse unmixing.

3.4 Proposed Algorithms

In this section, we present the proposed greedy algorithm, called OMP-Star\(^2\). A non-negative variant of this algorithm is developed later in this section. OMP-Star itself can be considered a variant of OMP with its abilities enhanced following an approach inspired by A-Star [85], hence named OMP-Star. Before giving a detailed account on the proposed algorithms, let us briefly discuss the most significant drawback of the greedy algorithms i.e. the problem of getting stuck in local optima of the solution space. This discussion will help in understanding the intuition behind the better performance of the proposed algorithms.

Consider the identification step of OMP, where it identifies the column \( d_{j_0} \) of \( D \), that minimizes the current residual. Assume another column of \( D \), \( d_k : k \neq j_0 \) that also causes almost the same amount of reduction in the residual, but it is not the minimum (because of \( d_{j_0} \)). Being greedy, OMP prefers \( d_{j_0} \) over \( d_k \) for augmenting

3.4. Proposed Algorithms

the selected subspace. Here, the question arises, should we simply neglect $d_k$ and make the locally optimal choice of $d_j$? What if, $d_k$ was actually the right vector to be picked but $d_j$ got selected only because of its similarity to $d_k$? Such locally optimal choices of greedy algorithms trap them in local optima - a problem well researched for greedy search algorithms in the Artificial Intelligence (AI) literature [178]. In the context of greedy search algorithms, A-Star [85] is among the most effective approaches to solve the problem of local optimality of the solutions. The main idea behind this approach is to make selections in the search procedure keeping in view a) the current benefits as well as b) the future benefits of making a selection. This futuristic greedy approach helps in avoiding the local optima.

3.4.1 OMP-Star

OMP-Star uses a futuristic approach in the greedy pursuit strategy. The proposed algorithm initializes by considering the complete signal $y$ as the residue. In the main iteration, it first identifies the column $d_{j_0}$ of $D$ that maximally correlates with the current residual vector. It uses this column to identify further $L$ columns in $D$ that best correlate with the residual vector. When normalized in $l_2$-norm, the inner product of each of these $L$ vectors with the residual vector is at least a fraction $t$ of the inner product of the unit vector of $d_{j_0}$ and the residual vector (line ‘7’ in Algorithm 2). Here, $t \in (0, 1)$ is an algorithm parameter. The indices of thus identified $L + 1$ columns form a set $T$. OMP-Star then points out the index $j^* \in T$ of the column that it chooses to augment the selected subspace. For a moment, let us skip the discussion on the selection procedure of $j^*$ (line ‘9’ to ‘13’ in Algorithm 2). Assume that $j^*$ is exactly the index we would like to select. This index is added to a set $S_{i-1}$ (i denotes the current iteration) containing the indices of $D$’s columns which form the selected subspace. Once the selected subspace is augmented with the column corresponding to $j^*$, its span is used for updating the residual vector like OMP. This process is repeated until one of the stopping rules in line ‘17’ of the algorithm is satisfied. Among these rules (a) and (b) are obvious and well known. In the rule (c) we use a residual decay parameter $\beta \in (0, 1)$. This rule ensures that the algorithm stops if it is not able to reduce the $l_2$-norm of the residual vector at least by a fraction $\beta$ in its last iteration.

For different iterations of OMP-Star, $|T|$ is expected to be different, where $|.|$ denotes the cardinality of the set. In a particular iteration, $|T|$ will be large if $D$ contains many columns with high correlation to $d_{j_0}$. On the other hand, if $d_{j_0}$ has very low correlation with other columns, $|T| = 1$. The parameter $t$ provides the
Algorithm 2 OMP-Star

Initialization:
1: Iteration: \( i = 0 \)
2: Initial solution: \( \alpha^0 = 0 \)
3: Initial residual: \( r^0 = y - D\alpha^0 = y \)
4: Selected support: \( S^0 = \text{support}\{\alpha^0\} = \emptyset \)

Main Iteration: Update iteration: \( i = i + 1 \)

Identification:
5: Compute \( \epsilon(j) = \min_{z_j} ||d_jz_j - r^{i-1}||_2^2, \forall j \in \{1, ..., k\} \), using the optimal choice
6: \( z^*_j = \frac{d_j^Tr^{i-1}}{||d_j||_2^2} \)
7: Find the minimizer, \( j_0 \) of \( \epsilon(j) : \forall j \notin S^{i-1}, \epsilon(j_0) \leq \epsilon(j) \)
8: Find \( j_1 \) to \( j_L : \frac{d_{j_1}^Tr^{i-1}}{||d_{j_1}||_2} \geq t \times \frac{d_{j_0}^Tr^{i-1}}{||d_{j_0}||_2} \)
9: If cardinality of \( T = 1 \) then
10: \( j^* = j_0 \)
11: else
12: \( j^* \leftarrow \text{ForwardSelection}(D, y, S^{i-1}, T, f) \)
13: end if

Augmentation:
14: \( S^i = S^{i-1} \cup \{j^*\} \)

Residual update:
15: \( \alpha^i = \min_{\alpha} ||D\alpha - y||_2^2 \) s.t. \( \text{support}\{\alpha^i\} = S^i \)
16: \( r^i = y - D\alpha^i \)

Stopping rule:
17: If a) \( i > \text{desired iterations}, \) or
b) \( ||r^i||_2 < \epsilon_0, \) or
c) \( ||r^i||_2 > \beta ||r^{i-1}||_2 \) stop,
otherwise iterate again.
3.4. Proposed Algorithms

Procedure ForwardSelection
Input: $D \in \mathbb{R}^{m \times k}$, $y \in \mathbb{R}^{m \times 1}$, $S$, $T$, $f$
Output: $j^*$

1: $R_0 = \emptyset$
2: for each element $T_i : i \in \{1, \ldots, z\}$ in $T$ do
3:   $S^0 = S \cup T_i$
4:   $\alpha^0 = (D_{S^0}^\top D_{S^0})^{-1} D_{S^0}^\top y : D_{S^0}$ = Matrix with columns of $D$ indexed in $S^0$
5:   $r^0 = y - D_{S^0} \alpha^0$
6: for $q = 1$ to $f$ do
7:   Compute $\epsilon(j) = \min_{z_j} \|d_j z_j - r^{q-1}\|_2^2$, $\forall j \in \{1, \ldots, k\}$, using $z_j^* = \frac{d_j r^{q-1}}{\|d_j\|_2}$
8:   Find the minimizer, $j_0$ of $\epsilon(j) : \forall j \notin S^{q-1}, \epsilon(j_0) \leq \epsilon(j)$
9:   $S^q = S^{q-1} \cup \{j_0\}$
10:  $\alpha^q = \min_{\alpha} \|D \alpha - y\|_2^2$ s.t. support$\{\alpha\} = S^q$
11:  $r^q = y - D \alpha^q$
12: end for

$R_i = R_{i-1} \cup (\sum_{\gamma=0}^{f} \|r^{\gamma}\|_2^2)$
13: end for
14: $j^*$ = Element of $T$ corresponding to the smallest element in $R_z$

The above process is repeated for each element of $T$. Once finished, the procedure finds the minimizer over the cumulative residuals. The element of $T$ corresponding to this minimizer is considered $j^*$.

Note that, the column corresponding to $j^*$ is a suitable choice in the current main iteration of OMP-Star because it is one of the columns that best correlate with the quantitative boundary for deciding on the high and low correlation. For the cases when $|T| = 1$, $j^*$ simply points to $d_{j_0}$ (line 9, 10 in Algorithm 2). Otherwise, the algorithm selects the index $j^*$ by calling out to a procedure ForwardSelection, (line ‘12’ in Algorithm 2). This procedure works as follows. It picks an element of $T$ and temporarily augments the selected subspace with the corresponding column of $D$. It then performs $f$ OMP-like iterations using this augmented subspace as the selected subspace. Each of these iterations may add a vector (i.e. a column of $D$) to the temporary subspace. Before the first iteration and after each iteration, the procedure notes the $l_2$-norm of the residual vector. Once $f$ iterations have been performed, it sums up the $f + 1$ values of the noted residuals and saves the cumulative residual. The last $f + 1$ columns of the temporarily augmented subspace are then removed. The above process is repeated for each element of $T$. Once finished, the procedure finds the minimizer over the cumulative residuals. The element of $T$ corresponding to this minimizer is considered $j^*$. 
residual. Among all the suitable choices, it is potentially the quickest in terms of reducing the residual. The later property is ensured by the ForwardSelection procedure that identifies the column by looking forward into f OMP-like iterations. This futuristic identification helps in mitigating the effects of the local optimality issue. This notion is similar to the main idea of A-Star, mentioned in Section 3.4. However, there is a discernible difference between the main objectives of the two approaches. Whereas the proposed approach only aims at improving the robustness against the local optima in the solution space, A-Star aims at always finding the globally optimal solution, usually at the cost of very high computation. A-Star uses a cost function that employs an admissible heuristic to ensure the global optimality of the solution. This solution is found by iteratively expanding the current best node of the search tree it parses. Therefore, A-Star keeps cost records of all the expanded nodes and backtracks when a previously expanded node has a cost lower than the currently expanded best node. This makes A-Star an optimal search algorithm, however its search strategy is computationally very expensive [178].

A*OMP proposed by Karahanoglu et al. [108] directly incorporates the idea of A-Star search into OMP algorithm. As mentioned in [108], similar to A-Star, the objective of A*OMP is to find the optimal solution without particularly considering the computational complexity. Therefore, the computation time of A*OMP is generally very high. Compromise over the computational complexity makes A*OMP less appealing for the problem of hyperspectral unmixing. In order to maintain the computational advantages of the greedy pursuit strategy, OMP-Star does not aim at the global optimality of the solution. However, it is able to mitigate the effects of local optimality by looking into its future iterations. Since A*OMP directly uses the A-Star search strategy, it also requires a cost function to decide on the best node to expand. Generally, combining this cost function with OMP is not straightforward [108]. An appropriately defined cost function also needs to compensate for the different path lengths during the search process, which adds to the complexity of this function. A*OMP uses path pruning strategy to make the search process tractable. Since the proposed algorithm does not directly integrate the A-Star search strategy with OMP, it does not suffer from these limitations. In comparison to A*OMP the proposed algorithm is simpler and computationally much more efficient.

3.4.2 OMP-Star+

Reflectances of materials are non-negative values. In other words, for the problem of sparse unmixing the columns of D will only have non-negative coefficients.
3.5. Coherence Reduction

Furthermore, \((P_0^{P+})\) in Equation (3.5) dictates that the coefficients of \(\alpha\) should always be non-negative. We make use of these constraints and further tailor OMP-Star for sparse unmixing. We denote this non-negative version of OMP-Star, as OMP-Star+. For conciseness, we only present the changes in OMP-Star that would convert it to OMP-Star+.

The first change must be made in line ‘5’ of Algorithm 2. In the \(i\)th main iteration, instead of simply minimizing \(||d_jz_j - r^{i-1}||_2^2\) over \(z_j\), we must also make sure that the minimizers are non-negative quantities. Mathematically, \(\forall j \in \{1, ..., k\}\)

\[
\epsilon(j) = \min_{z_j \geq 0} ||d_jz_j - r^{i-1}||_2^2
\]

\[
= ||r^{i-1}||_2^2 - \max\{d_j^Tr^{i-1}, 0\}^2 \frac{||d_j||_2^2}{||d_j||_2^2}.
\]

Computing \(\epsilon(j)\) as above implies that all the vectors corresponding to the set \(T\) (line ‘8’ in Algorithm 2) will be almost parallel to the residual vector \(r^{i-1}\). This is different from the case of OMP-Star, where we also allow the vectors to be almost anti-parallel to \(r^{i-1}\). Since OMP-Star can use negative elements in \(\alpha\), it can later invert the direction of the suitable vectors. However, in OMP-Star+ it is not possible because here we constrain the elements of \(\alpha\) to be positive. This constraint is imposed by making a second change in Algorithm 2. We change the optimization problem in line ‘15’ of the Algorithm with the following constrained optimization problem:

\[
\alpha^i = \min_{\alpha} ||D\alpha - y||_2^2, \text{ s.t. } \text{support}\{\alpha^i\} = S^i, \alpha \geq 0.
\]

The above changes impose non-negativity in the main iteration of OMP-Star. We also need to impose this constraint in the ForwardSelection procedure in order to fully convert OMP-Star to OMP-Star+. Thus, for OMP-Star+, we compute \(\epsilon(j)\) (line ‘7’ of the Procedure) using the following equation:

\[
\epsilon(j) = ||r^{q-1}||_2^2 - \max\{d_j^Tr^{q-1}, 0\}^2 \frac{||d_j||_2^2}{||d_j||_2^2}, \forall j \in \{1, ..., k\}.
\]

Similarly, the optimization problem in line ‘10’ of the Procedure is replaced with the corresponding constrained optimization problem. These changes ensure that the selection of \(j^*\) made by the procedure follow the non-negativity constraint.

3.5 Coherence Reduction

In sparse approximation problems, coherence \(\mu\) is one of the most fundamental characteristics associated with \(D\) [196]. The coherence of a matrix denotes the
maximum absolute inner product between any two distinct columns, as follows:

$$\mu = \max_{i,j \neq i} \frac{|d_i^T d_j|}{||d_i||_2 ||d_j||_2}. \quad (3.13)$$

Consider the following noiseless SA problem:

$$\begin{align*}
(P_0) : \quad & \min_{\alpha} ||\alpha||_0 \quad s.t. \quad D\alpha = y. \quad (3.14)
\end{align*}$$

If \(y\) is a linear combination of \(p\) distinct columns of \(D\), then we are interested in the upper bound on \(p\) such that the support of the solution \(\alpha\) always points to the correct \(p\) columns of \(D\). Tropp [196] showed that this bound depends on \(\mu\):

$$p < \frac{1}{2} \left( 1 + \frac{1}{\mu} \right). \quad (3.15)$$

The bound has been proven in [196] for OMP as well as BP. This bound is also valid for the algorithms proposed in this work\(^3\). Elad [31] modified the bound by replacing \(\mu\) in (3.15) with \(\mu_{\Delta}\), where \(\mu_{\Delta}\) is the coherence of \(PD\). Here, \(P\) must be a square invertible matrix. This modification is achieved by replacing the constraint in \((P_0)\) by \(PD\alpha = Py\).

Note that the bound in (3.15) is pessimistic because \(\mu\) is the worst case measure. If the columns of \(D\) have high correlation, the inequality (3.15) becomes practically meaningless. For instance, according to the above inequality two mixed signals in \(y\) (i.e. \(p = 2\)) requires \(\mu < 0.33\) for their guaranteed exact identification. However, \(\mu \approx 1\) when columns of \(D\) have high correlation, as in sparse unmixing [102]. Therefore, the inequality (3.15) is not meaningful for sparse unmixing. However, sparse unmixing can benefit from two important observations based on the above discussion. 1) Low correlation among the columns of \(D\) favors accurate support of the solution. 2) An invertible matrix \(P\) can modify this correlation without affecting the results. Based on these observations, we focus on finding the invertible matrix \(P\) which reduces the correlation among the columns of \(D\), thereby improving the accuracy of the solution support.

Tsai and Philpot [200] used derivatives to analyze remote sensing data with high spectral resolution. Bieniarz et al. [19] recently showed that, generally, the correlation among spectra can be reduced with the help of derivatives. Therefore, we use the same tool to construct the desired matrix \(P\) for the hyperspectral unmixing

\(^3\)The futuristic greedy heuristic will give the same result as the greedy heuristic, if the latter is guaranteed to find the optimal result. This happens when the data satisfies the inequality (3.15).
3.5. Coherence Reduction

problem. For a spectra captured at constant wavelength intervals, we can formulate the derivative operator \((\Delta)\) in its generic form as below:

\[
\Delta^o d_b = \frac{1}{(\Delta \lambda)^o} \sum_{i=0}^{o} (-1)^i \binom{o}{i} d_{b+[o-i]s},
\]

where \(d_b\) is the reflectance of the spectra at band \(b\). Among the parameters of the operator, \(o\) is the order of the derivative and \(s\) is the band difference. The band difference denotes the number of bands over which the slope is calculated. Domains of both of the aforementioned parameters are positive integers. On the right hand side, \(\Delta \lambda\) denotes the wavelength difference between the bands over which the slope is computed.

We exemplify the construction of \(P\) from Equation (3.16) for particular values of \(s\) and \(o\). Consider a spectra given as a vector \(d \in \mathbb{R}^m\), for which the wavelength difference between two consecutive samples is 0.01\(\mu m\). If we choose \(o = 1\) and \(s = 2\) in Equation (3.16), the equation simplifies to:

\[
\Delta^1_2 d_b = \frac{d_{b+2} - d_b}{2 \times 10^{-9}}.
\]

By definition, the above derivative can be computed for each band of \(d\) except the last \(s = 2\) bands. For those bands the derivative operation keeps the original reflectance values of the signal. \(\Delta^1_2\) operates band-by-band. It is also possible to take the derivative of the complete vector \(d\) by left multiplying it with \(\gamma P\), where \(\gamma \in \mathbb{R}\) is a constant and \(P\) is a matrix in \(\mathbb{R}^{m \times m}\). For \(\Delta^1_2\), \(\gamma = -1/(2 \times 10^{-9})\) and \(P\) is a matrix with its elements \(p_{i,i} = 1, \forall i \in \{1, ..., m\}\) and \(p_{i,i+2} = -1, \forall i \in \{1, ..., m-2\}\). All other elements of \(P\) are zero.

The square matrix \(P\) created for \(\Delta^1_2\) is an invertible matrix, because it has 1’s at its diagonal and non-zero elements only in the upper triangle. In fact, this property holds for all the \(P\) matrices created with the above strategy for any values of \(s\) and \(o\) satisfying the inequality\(^4\) \((1 + os) \leq m\). The \(P\) matrix thus created can directly be left multiplied with \(D\) to take derivatives of all the spectra in the library simultaneously. We can either choose to ignore \(\gamma\) or multiply it with \(P\) and use this modified matrix. We prefer the latter choice as it corresponds to the exact definition of derivatives used in [200]. In general, the derivative operator is able to enhance the dissimilarities between two seemingly identical spectra. This results in the reduction of correlation between the spectra.

\(^4\)The inequality follows from the general definition of the derivative given in Equation (3.16) and gives the upper bound on the parameter values.
3.6 Experiments with Synthetic Data

In this section we evaluate the proposed approach with synthetic data. The experiments performed on synthetic data are significant because the endmembers and their corresponding fractional abundances are known, enabling quantitative analysis of the unmixing. After introducing the data used in the experiments, we first discuss the preprocessing of the data with derivatives. This is followed by the experiments for evaluating the proposed algorithms for endmemember identification and fractional abundance estimation.

3.6.1 Data

In the experiments with synthetic data, we created our matrix $D$ from the NASA Jet Propulsion Laboratory’s Advanced Space-borne Thermal Emission and Reflectance Radiometer (ASTER) Library\(^5\). This library contains pure spectra of 2400 natural and man-made materials. For the experiments, we chose 425 of these spectra. The spectra were chosen such that their coherence $\mu$ is 0.9986. We kept $\mu < 1$ in order to ensure that all the selected spectra were unique. We resampled the spectra in the wavelength range 0.4 to 2.5$\mu$m, at a constant interval of 10$nm$. This resulted in 224 samples per spectra. We performed this resampling in order to match the sampling strategy of NASA’s Airborne Visible Infrared Imaging Spectrometer (AVIRIS) [80]. From the resampled spectra we dropped first 24 bands because of zero or very low reflectance values. We created $D$ with the processed spectra, which made it a $200 \times 425$ matrix. Henceforth, we use the term ‘library’ for $D$. All the techniques compared in the experiments use the same library. Among the 425 spectra in the library, 285 belong to minerals, 15 belong to soils and 125 represent rocks.

For the unmixing problem, we create a mixed signal/pixel $y$ by randomly selecting $p$ spectra from the library. We simulate the hyperspectral data cube as 500 mixed pixels, in which each pixel comprises $p$ different randomly selected spectra/endmembers. Thus, we simulate a highly mixed scenario. Pixel-based algorithms enjoy good performance in these scenarios because they do not depend on the contextual or spatial information in the data. Since the proposed algorithms are pixel-based, we compare their performances only with the other pixel-based algorithms. Following the experimental protocol in [102], we draw the fractional abundances of the endmembers in each pixel of the hyperspectral data cube from a Dirichlet distribution. Therefore, the fractional abundances in each pixel sum up to

\(^5\)Available at http://speclib.jpl.nasa.gov.
3.6. Experiments with Synthetic Data

1 and ASC holds for each pixel. Unless stated otherwise, the results reported below are the mean values computed over the hyperspectral data cube.

3.6.2 Analysis of Derivatives

To analyze the correlation reduction characteristic of derivatives, we introduce the mean coherence \( \bar{\mu} \) for a spectral library \( D \)

\[
\bar{\mu} = \frac{1}{k} \left( \sum_{i=1; i \neq j}^{k} \max_{j} \frac{|d_i^T d_j|}{||d_i||_2||d_j||_2} \right),
\]

which maximizes the correlation of each spectra \( d_i \) over all the other spectra in the library and takes the mean of the computed maximum values. Note that, spectral derivatives generally reduce the correlation among the spectra. Since, \( \mu \) is only the worst case measure, it is not possible to correctly analyze this property of the spectral derivatives using \( \mu \). Compared to \( \mu \), \( \bar{\mu} \) gives a better picture of the similarity among the spectra in the library because every spectra in \( D \) contributes to \( \bar{\mu} \). The mean coherence of \( D \) in our experiments is 0.9974. We calculated the difference between the mean coherences of \( D \) before taking its derivative (i.e. \( \bar{\mu} \)) and after taking the derivative (i.e. \( \bar{\mu}_\Delta \)). The greater is this value, the more advantageous is the derivative operation for the sparse unmixing problem. Fig. 3.2a shows the plots for \( \bar{\mu} - \bar{\mu}_\Delta \) (scaled by a factor of 100) as a function of band difference \( s \). The plots are shown for the 1\textsuperscript{st}, 2\textsuperscript{nd} and 3\textsuperscript{rd} order derivatives.

In order to benefit from derivatives in sparse unmixing, they must also be operated on the mixed pixels. The mixed pixels can be noisy, whereas a derivative operation is usually sensitive to noise. This is because, derivatives amplify the sudden changes in the reflectance patterns of spectra. Keeping in view the noise sensitivity of derivatives, we analyzed the effects of noise amplification caused by derivatives. For this purpose, we corrupted each spectra in \( D \) with additive noise to create a library \( D_\Delta \). Each spectra in \( D_\Delta \) has SNR = 35dB. We took the derivatives of \( D_\Delta \) with different values for the parameters \( s \) and \( o \). Each time, we computed SNR\( _\Delta \) using the following formula:

\[
SNR_\Delta = 10 \log_{10} \left( \frac{E[||d_\Delta||_2^2]}{E[||d_\Delta - d_\Delta||_2^2]} \right).
\]

The above expression estimates the SNR of the differentiated library \( D_\Delta \) in decibel. In this expression, \( d_\Delta \) denotes the spectra in the differentiated library \( D_\Delta \), \( d_\Delta \) denotes the spectra in \( D_\Delta \) and \( E[\cdot] \) is the expectation operator. In our experiments, we have used white noise as well as correlated noise. Therefore, we
Figure 3.2: Analysis of derivatives: (a) Difference between the mean coherence of the library before and after taking the derivative (scaled by a factor of 100), as a function of the band difference $s$. (b) Difference between SNR and SNR$_\Delta$, as a function of $s$ for white noise. (c) Difference between SNR and SNR$_\Delta$, as a function of $s$ for correlated noise. SNR = $35\text{dB}$ in (b) and (c).

separately analyze their amplification with derivatives. We have followed Bioucas-Dias and Nascimento [23] to generate both types of the additive noise. Let $R_n = \text{diag}(\sigma_1^2, ..., \sigma_m^2)$ be the correlation matrix of the additive noise, where the diagonal of the matrix follows a Gaussian shape centered at the band $m/2$ and

$$
\sigma_i^2 = \frac{\sigma^2}{\eta \sqrt{2\pi}} e^{-\frac{(i-m)^2}{2\eta^2}} \forall i \in \{1, ..., m\}.
$$

(3.20)

In (3.20), the parameter $\eta$ controls the variance of the Gaussian shape (i.e. $\eta \to 0$ generates a single band noise and $\eta \to \infty$ results in white noise), whereas,

$$
\sigma^2 = \frac{\sum_{j=1}^{k} ||d_j||^2}{k \times 10^{-\left(\frac{3\text{SNR}}{10}\right)}}
$$

(3.21)

controls the total noise power. Further details on the above mentioned noise model can be found in [23]. For the fixed value of SNR = $35\text{dB}$, we vary the value of $\eta$ to obtain the desired type of noise. In all of our experiments, we have used $\eta = 500$. 

3.6. Experiments with Synthetic Data

Figure 3.3: Example of white and correlated noise: In the given original reflectance spectra of a mineral, white and correlated noise have been added to obtain SNR = 35dB. The white noise is generated by choosing $\eta = 500$, whereas $\eta = 18$ for the correlated noise. Band 50 to 150 are corrupted with the correlated noise, with maximum noise power around band 100.

Fig. 3.2b and 3.2c plot the difference in SNR and $\text{SNR}_\Delta$, as a function of the band difference $s$ for different values of $o$, for white and correlated noise respectively. The plots in the figures remain qualitatively similar for different values of SNR (in the range 20 to 100dB). We restricted the analysis only up to 3rd order derivatives in Fig. 3.2 because the noise amplification for higher order derivatives becomes too large, rendering the derivative operation unbenefficial for sparse unmixing.

Based on the plots in Fig. 3.2, we selected $s = 5$ and $o = 1$ for the derivative operation in our experiments, presented below. These values of the parameters provide a reasonable tradeoff between the correlation reduction and noise amplification at SNR = 35dB. Notice that, the effect of noise amplification reduces with the increase in the value of $s$. This is due to the smoothing caused by the derivative operation, which is illustrated in Fig. 3.4. In general, for small values of SNR it is more beneficial to use larger values of $s$ to control the noise amplification, whereas for high SNR of the spectral signatures $s$ should be kept small (e.g. 1 or 2) to fully exploit the coherence reduction ability of derivatives.

3.6.3 Experiments for Endmembers Identification

Unmixing Fidelity: A hyperspectral unmixing approach should be evaluated according to the two goals of unmixing mentioned in Section 3.2.2. For the first goal of ‘finding the correct set of endmembers’, we introduce the evaluation metric of unmixing fidelity of a sparse unmixing solution, $\Phi(\alpha) \to [0,1]$. If $\mathcal{P} = \{x \mid x$ is the index of an endmember in $D\}$ and $\mathcal{A} = \{a \mid a$ is a non-zero elements in $\alpha\}$, then:
Figure 3.4: Smoothing effect by increasing $s$ in derivatives: (a) Spectra of a mineral. (b) Solid line: derivative of the pure spectra. Dotted line: derivative of the noisy spectra. Derivatives taken with $o = 1$ and $s = 1$. (c) Derivatives taken with $o = 1$ and $s = 2$. The resulting plots are smoother and the noise amplification has reduced.

\[ \Phi(\alpha) = \frac{|P \cap A|}{|A|}, \]  

(3.22)

here, $|.|$ denotes the cardinality of the set. Unmixing fidelity calculates the fraction of the correct support in the solution $\alpha$. It does not take into account the fractional abundances because, with known correct support of $\alpha$, it is always possible to compute the fractional abundances separately (e.g., using least squares).

**Parameters Values Selection:** Before experimenting with the proposed algorithms, we must select the values of the parameters $t$ and $f$ for the algorithms. In an iteration, $t$ decides the columns of $D$ to be chosen for the ForwardSelection procedure, whereas $f$ is the number of OMP-like iterations performed by the procedure to select the best column. We selected the values of these parameters by experimenting on a separate training data, created with the same protocol as mentioned in Section 3.6.1. In these experiments, the algorithms were given the library and the pixel after operating $\Delta_5^1$ on them. For the library, the derivative was operated
after normalizing its columns in $l_1$-norm. Note that, we follow the same procedure of preprocessing the data for all the greedy algorithms in the experiments to follow. For the selection of values of parameters, we chose the data to be noise-free, with $p = 5$ for each pixel. We analyzed the unmixing results for different values of the parameters for the proposed algorithms. These parameters offer a trade-off between $\Phi$ and the computation time of the algorithms. Fig. 3.5 shows plots of the results that also offer insights into the general behavior of the algorithms for different values of the parameters.

Fig. 3.5a shows that larger values of $t$ reduce the unmixing fidelity of the solution. However, they also decrease the computation time of the algorithms in a linear fashion, as shown in Fig. 3.5b. This trend is the same for OMP-Star and OMP-Star+. Fig. 3.5c shows the effects of increasing $f$ on $\Phi$. In the figure, the three groups of curves correspond to three different values of $p$. It is clear from the figure that increasing $f$ generally improves the unmixing fidelity of the solution. This statement is valid for all the values of $p$. Notice that, with an increase in $p$ the...
unmixing fidelity decreases. This is a general trend for sparse unmixing algorithms. However, the fact that the cardinality of a mixed pixel in practical scenarios is usually of the order of four to five [102], mitigates the adverse effects of this trend. For the sake of thoroughness of the analysis, we used values of \( p \) as high as 10 in our experiments.

**Results:** Based on the above mentioned experiments we selected \( t = 0.92 \) and \( f = 2 \) for OMP-Star as well as OMP-Star+. We have performed experiments for the comparison of these algorithms with all the other greedy pursuit algorithms mentioned in Section 3.3. Fig. 3.6 shows this comparison in terms of unmixing fidelity of the solutions as a function of \( p \). The experiments were performed for the cases when the mixed pixels were noise-free (Fig. 3.6a) and noisy. For the noisy case, we chose \( \text{SNR} = 35 dB \) with additive white noise (Fig. 3.6b) and correlated noise (Fig. 3.6c).

In Fig. 3.6, the algorithms represented by dotted lines are impractical choices for hyperspectral unmixing, however we have included their results for the sake of thoroughness of the analysis. Among these algorithms SP, CoSaMP and ROMP must know the value of \( p \) a priori. In hyperspectral unmixing, this value is usually unknown. We have also shown LAOMP and A\(^*\)OMP with dotted lines because their computational time is so high that they loose the computational advantages of the greedy approach (see Section 3.8). A\(^*\)OMP is one algorithm that is consistently able to show better results than the proposed algorithm for \( p \geq 5 \). However, the proposed algorithms have computed these results nearly 200 times faster than A\(^*\)OMP. It is also worth mentioning here that \( p \) is usually 4 to 5 in practical scenarios [102].

For the experiments, the data has been pre-processed with spectral derivatives for all the greedy algorithms, including the proposed ones. As can be seen in Fig. 3.2a, the pre-processing reduces the coherence between the spectral signatures. However, this reduction is not sufficient to achieve accurate endmember identification using the greedy algorithms. Therefore, even after the pre-processing of the data, a greedy algorithm must show robustness to the high coherence of the data for better results. In the proposed algorithms, this robustness is achieved by the futuristic heuristic, which resulted in the better performance.

In this section we have assumed that \( p \) is known beforehand (the assumption will be removed in the next section). Therefore, the proposed algorithms use stopping rule (a) in line ‘17’ of Algorithm 2 (Section 3.4). All the other algorithms also use the same stopping rule in this analysis. For LAOMP, we have selected \( L = 3 \) and for gOMP, we have chosen \( L = 2 \). Correct values of \( p \) have been given to OMP,
3.6. Experiments with Synthetic Data

Figure 3.6: Performance comparison of greedy algorithms: The plots show unmixing fidelity of the solutions found by different algorithms as a function of cardinality of the mixed pixels. (a) Noiseless case. (b) 35 dB SNR with white noise. (c) 35 dB SNR with correlated noise.

OMP+, ROMP, SP, CoSaMP, LAOMP and gOMP as their input parameters. For $A^*\text{OMP}$ we have chosen $I = 3$, $B = 40$, $\beta = 1.2$, $Kmax = 10$ and $\epsilon = 10^{-7}$. These parameters respectively represent the ‘number of initial search paths’, ‘the branching factor in each iteration’, ‘the parameter of adaptive auxiliary function’, ‘maximum allowed sparsity level’ and ‘the residual power for termination of the algorithm’. We have chosen the ‘adaptive’ cost function for $A^*\text{OMP}$ (see [108]) in all of our experiments, as this function resulted in the best performance of the algorithm. For each of the algorithm, we have optimized the parameter values using separate training data. With the above parameter settings, we have also analyzed the algorithms for the unmixing fidelity as a function of SNR of the mixed signals, with $p = 5$. Results of these experiments are shown in Fig. 3.7. It can be seen from the figure that the proposed algorithms are able to consistently perform well.
Chapter 3. Futuristic Greedy Approach to Sparse Unmixing

3.6.4 Experiments for Fractional Abundance

Once the endmembers have been identified, the unmixing problem reduces to computing the fractional abundances of the identified endmembers. If $\alpha \in \mathbb{R}^k$ is the estimated fractional abundance vector and $\alpha_\circ \in \mathbb{R}^k$ is the vector containing the actual fractional abundances, then the Euclidean distance $||\alpha_\circ - \alpha||_2$ is a suitable metric for evaluating the solution for fractional abundance estimation. Fig. 3.8 uses this metric to compare the performance of the proposed algorithms against the popular unmixing algorithms which are based on convex relaxation of the problem. Among these, SUnSAL+ and CSUnSAL+ are the non-negative variants of SUnSAL and CSUnSAL. Fig. 3.8 shows that the performance of the proposed algorithms is comparable to these well known algorithms. We should mention that the proposed algorithms achieve these results much faster than the other algorithms shown in the figure. Comparison of execution times of all the algorithms is given in Section 3.8.

For the proposed algorithms, the results shown in Fig. 3.8 are obtained using the following strategy:

1. Compute $\alpha$ after taking the derivative of the data, as mentioned in Section 3.6.3.

2. Create $D_A$ from the columns of $D$ corresponding to the support of $\alpha$.

3. Use $D_A$ and $y$ as inputs for CSUnSAL+, which calculates the fractional abundances.

4. Replace the fractional abundances in $\alpha$ with the corresponding new values.

Figure 3.7: Unmixing fidelity as a function of SNR: For all the plots $p = 5$. (a) White noise. (b) Correlated noise.
3.6. Experiments with Synthetic Data

Figure 3.8: Euclidean distance between the original fractional abundances vector and the computed fractional abundances vector: (a) 35 dB SNR with white noise. (b) 35 dB SNR with correlated noise.

The above strategy is similar to the library pruning concept of [183], but we use the algorithm CSUnSAL+, only for estimating the fractional abundances. In Fig. 3.8, the value of the regularization parameter $\lambda$ (see (3.7), Section 3.2.2) is set to $10^{-5}$ for all of the convex relaxation based algorithms, whereas the tolerance for the primal and dual residuals (see [22] for details) has been set to $10^{-3}$ for SUnSAL and CSUnSAL, and $10^{-5}$ for SUnSAL+ and CSUnSAL+. All of these values have been optimized on separate training data. We have used the same parameter settings for CSUnSAL+ when it is used in the above mentioned strategy. For the convex relaxation algorithms we have reported the results on the original data. When differentiated data is used with these algorithms the amplified noise causes problems in accurate estimation of abundances. In the presence of high noise, setting the algorithms for high fidelity signal reconstruction leads to poor estimates of abundances because the computed abundances also approximate the dominant noise component. To reduce this effect we can set the algorithms with high tolerance for signal reconstruction. However, it does not solve the problem completely. Due to high tolerance, the underlying original signal is also poorly approximated by the estimated abundances, resulting in compromised abundance estimation. Furthermore, increasing the tolerance beyond a certain range drastically aggravates the accuracy of the computed abundances. Therefore, choosing the right tolerance value becomes non-trivial. We avoid this issue in the case of the greedy algorithms because of the above mentioned strategy. Since the abundances are finally computed on the original data (and not on the differentiated data) in step (4), we do not encounter the problems caused by the noise amplification. We have also
experimented with the convex relaxation based algorithms using the differentiated data. However, despite careful optimization of the parameters, the abundances estimated by these algorithms using the differentiated data were less accurate as compared to those computed with the original data. Therefore, we have reported the results on the original data in Fig. 3.8. Notice that, we have also not included the convex relaxation based algorithms in the comparison for unmixing fidelity in Section 3.6.3. This is because, their results are not comparable with the proposed algorithms for that metric.

For instance, Fig. 3.9 shows two examples of the solutions found by OMP-Star+ and CSUnSAL+. In both cases, for the five actual endmembers (first row), OMP-Star+ identifies five endmembers (second row), out of which four are correctly identified. This results in $\Phi(\alpha) = 0.8$ in both cases. On the other hand, CSUnSAL+ identifies several endmembers with non-zero fractional abundances. In Fig. 3.9a, noting that many of the non-zero abundances are very small, we can set a threshold below which the abundances can be neglected. This can improve the unmixing fidelity of the solution. However, choosing the value of the threshold becomes tricky in the cases similar to Fig. 3.9b, where some correctly identified endmembers have fractional abundances close to or smaller than the wrongly identified endmembers. In Fig. 3.9b, choosing 0.05 as a threshold would result in the detection of 8 endmembers out of which only 4 would be correct, whereas a value of 0.1 would give only two endmembers.

For the results shown in Fig. 3.8 and 3.9, we did not assume prior knowledge of the cardinalities of the mixed pixels. Therefore, the proposed algorithms use stopping rule (c) in line ‘17’ of Algorithm 2, with $\beta = 0.9$. The value of the residual decay parameter $\beta$ was chosen with the help of separate training data. In order to compare the proposed algorithms with the other greedy algorithms for fractional abundance estimation, we again assumed that the value of $p$ is known for these algorithms (the proposed algorithms do not use the value of $p$ here as well). In this case, for all the greedy algorithms, including the proposed algorithms, we computed the fractional abundances following the above mentioned strategy with a small variation in step (3). Instead of using CSUnSAL+ in this step, we used the non-negative least squares method to estimate the fractional abundances. The results are shown in Table 3.1 for the data with Gaussian white noise and in Table 3.2 for the data with correlated noise. To obtain these results, we normalized $D_A$ and $y$ in $l_1$-norm before computing the fractional abundances. According to the analytical results of theorem (3.1) in Section 3.2.3, this automatically imposes ASC along the non-negativity
3.7 Experiments with Real Data

One interesting observation in these tables is that for $p \geq 5$, some of the algorithms perform slightly better than the proposed algorithms. This is especially true for gOMP. However, gOMP’s performance for unmixing fidelity is the poorest among the greedy algorithms (see Fig. 3.6 and 3.7). The reason for this phenomenon is simple. For the large values of $p$, the average true fractional abundance of the constraint over the least squares method. Here, we have taken advantage of the results of the theorem only in computing the fractional abundances. Generally, the process of differentiation can introduce some negative values in a non-negative matrix. However, since $D_A$ and $y$ contain the undifferentiated signals, the results of the theorem hold for this data. In all the experiments in this work, we have used the differentiated data only in identifying the correct set of the endmembers of the mixed pixels using the greedy algorithms (including the proposed algorithms). The fractional abundances are always computed using the original spectral signatures. Therefore, we have been able to take advantage of the results of theorem (3.1) for all the greedy algorithms. It is worth mentioning that spectral derivatives have never been exploited for greedy algorithms earlier in this manner.

3.7 Experiments with Real Data

Figure 3.9: Examples of fractional abundances estimated by OMP-Star+ and CSUnSAL+: The horizontal axis shows the indices of spectra in $D$. (First row) Actual fractional abundances of the 5 endmembers. (Second row) Fractional abundances of the endmembers computed by OMP-Star+. (Third row) Fractional abundances of the endmembers computed by CSUnSAL+. $||\alpha_0 - \alpha||_2$ for OMP-STAR+ and CSUnSAL+ are 0.35 and 0.20 in (a) and 0.42 and 0.30 in (b).
Table 3.1: Comparison of greedy algorithms for fractional abundance estimation: Below each algorithm’s name, the column shows the values of $||\mathbf{a}_0 - \mathbf{a}||_2$ for the cardinality of the mixed pixels in the first column. SNR = 35dB for the mixed signals with white noise.

<table>
<thead>
<tr>
<th>Card.</th>
<th>OMP</th>
<th>OMP+</th>
<th>OMP-Star</th>
<th>OMP-Star+</th>
<th>ROMP</th>
<th>LAOMP</th>
<th>SP</th>
<th>CoSaMP</th>
<th>gOMP</th>
<th>A^*OMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.413</td>
<td>0.408</td>
<td>0.311</td>
<td>0.305</td>
<td>0.359</td>
<td>0.351</td>
<td>0.393</td>
<td>0.476</td>
<td>0.359</td>
<td>0.501</td>
</tr>
<tr>
<td>3</td>
<td>0.508</td>
<td>0.498</td>
<td>0.402</td>
<td>0.392</td>
<td>0.511</td>
<td>0.438</td>
<td>0.517</td>
<td>0.586</td>
<td>0.421</td>
<td>0.493</td>
</tr>
<tr>
<td>4</td>
<td>0.564</td>
<td>0.552</td>
<td>0.461</td>
<td>0.456</td>
<td>0.560</td>
<td>0.485</td>
<td>0.560</td>
<td>0.632</td>
<td>0.470</td>
<td>0.485</td>
</tr>
<tr>
<td>5</td>
<td>0.580</td>
<td>0.552</td>
<td>0.521</td>
<td>0.504</td>
<td>0.573</td>
<td>0.480</td>
<td>0.582</td>
<td>0.661</td>
<td>0.496</td>
<td>0.480</td>
</tr>
<tr>
<td>6</td>
<td>0.573</td>
<td>0.561</td>
<td>0.530</td>
<td>0.518</td>
<td>0.556</td>
<td>0.512</td>
<td>0.584</td>
<td>0.671</td>
<td>0.488</td>
<td>0.467</td>
</tr>
<tr>
<td>7</td>
<td>0.577</td>
<td>0.541</td>
<td>0.545</td>
<td>0.530</td>
<td>0.545</td>
<td>0.507</td>
<td>0.584</td>
<td>0.676</td>
<td>0.490</td>
<td>0.451</td>
</tr>
<tr>
<td>8</td>
<td>0.575</td>
<td>0.539</td>
<td>0.561</td>
<td>0.551</td>
<td>0.535</td>
<td>0.522</td>
<td>0.586</td>
<td>0.681</td>
<td>0.488</td>
<td>0.449</td>
</tr>
<tr>
<td>9</td>
<td>0.581</td>
<td>0.541</td>
<td>0.571</td>
<td>0.562</td>
<td>0.527</td>
<td>0.511</td>
<td>0.581</td>
<td>0.673</td>
<td>0.487</td>
<td>0.453</td>
</tr>
<tr>
<td>10</td>
<td>0.591</td>
<td>0.532</td>
<td>0.579</td>
<td>0.569</td>
<td>0.519</td>
<td>0.497</td>
<td>0.586</td>
<td>0.661</td>
<td>0.489</td>
<td>0.459</td>
</tr>
</tbody>
</table>

Table 3.2: Experiments of Table 3.1, repeated with correlated noise.

<table>
<thead>
<tr>
<th>Card.</th>
<th>OMP</th>
<th>OMP+</th>
<th>OMP-Star</th>
<th>OMP-Star+</th>
<th>ROMP</th>
<th>LAOMP</th>
<th>SP</th>
<th>CoSaMP</th>
<th>gOMP</th>
<th>A^*OMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.406</td>
<td>0.405</td>
<td>0.328</td>
<td>0.324</td>
<td>0.355</td>
<td>0.348</td>
<td>0.381</td>
<td>0.448</td>
<td>0.355</td>
<td>0.398</td>
</tr>
<tr>
<td>3</td>
<td>0.503</td>
<td>0.495</td>
<td>0.378</td>
<td>0.370</td>
<td>0.539</td>
<td>0.392</td>
<td>0.479</td>
<td>0.501</td>
<td>0.417</td>
<td>0.386</td>
</tr>
<tr>
<td>4</td>
<td>0.556</td>
<td>0.541</td>
<td>0.423</td>
<td>0.404</td>
<td>0.582</td>
<td>0.437</td>
<td>0.534</td>
<td>0.589</td>
<td>0.462</td>
<td>0.379</td>
</tr>
<tr>
<td>5</td>
<td>0.571</td>
<td>0.544</td>
<td>0.460</td>
<td>0.433</td>
<td>0.582</td>
<td>0.449</td>
<td>0.554</td>
<td>0.604</td>
<td>0.485</td>
<td>0.370</td>
</tr>
<tr>
<td>6</td>
<td>0.551</td>
<td>0.521</td>
<td>0.466</td>
<td>0.433</td>
<td>0.559</td>
<td>0.457</td>
<td>0.524</td>
<td>0.598</td>
<td>0.455</td>
<td>0.372</td>
</tr>
<tr>
<td>7</td>
<td>0.551</td>
<td>0.517</td>
<td>0.468</td>
<td>0.450</td>
<td>0.547</td>
<td>0.451</td>
<td>0.534</td>
<td>0.610</td>
<td>0.457</td>
<td>0.369</td>
</tr>
<tr>
<td>8</td>
<td>0.537</td>
<td>0.498</td>
<td>0.481</td>
<td>0.447</td>
<td>0.527</td>
<td>0.444</td>
<td>0.518</td>
<td>0.602</td>
<td>0.449</td>
<td>0.376</td>
</tr>
<tr>
<td>9</td>
<td>0.532</td>
<td>0.488</td>
<td>0.501</td>
<td>0.469</td>
<td>0.526</td>
<td>0.439</td>
<td>0.529</td>
<td>0.609</td>
<td>0.450</td>
<td>0.382</td>
</tr>
<tr>
<td>10</td>
<td>0.529</td>
<td>0.486</td>
<td>0.502</td>
<td>0.467</td>
<td>0.507</td>
<td>0.443</td>
<td>0.520</td>
<td>0.613</td>
<td>0.450</td>
<td>0.390</td>
</tr>
</tbody>
</table>

endmembers becomes very small for a pixel. Algorithms that over estimate the number of endmembers in a pixel (e.g., gOMP) generally assign small values to all the fractional abundances because of ASC. For the correctly identified endmembers, small values become good approximations of the actual fractional abundances. At the same time, small fractional abundances are also good approximations for the falsely identified endmembers (as their true fractional abundances are zero). This results in an overall reduction of $||\mathbf{a}_0 - \mathbf{a}||_2$. The algorithms that do not over estimate the number of endmembers, can get heavily penalized when they detect wrong endmembers when $p$ is high. This is because, they end up assigning relatively large values to the fractional abundances when the actual values are zero. Despite this, in
our opinion, $||\alpha_0 - \alpha||_2$ is still a good metric for evaluation of fractional abundance estimations as it accounts for all the endmembers identified by an algorithm. This metric gives a clear picture of the results, especially for $p \leq 5$ (which corresponds to practical scenarios). The root mean square error (RMSE) metric used in [162] and [183] is evaluated with only the correctly identified endmembers. Therefore, it generally favors the over estimating algorithms for all the values of $p$.

In this section we evaluate the proposed algorithms on real-world hyperspectral data. We compute the abundance maps of the materials for the well known AVIRIS Cuprite data set. This data set was collected over a region in Cuprite mining district Nevada. The mineral classification map for this region, made available by the US Geological Survey, is shown in Fig. 3.10. The classification map shown in the figure is computed with Tricorder software product. In our experiments, we use the hyperspectral data cube corresponding to the region inside the rectangle. We use 188 spectral bands of this data set (low SNR bands were dropped), thus, the used hyperspectral cube has the dimensions $350 \times 350 \times 188$. In these experiments we create our library ($D^{188\times325}$) from the USGS spectral library [55], that includes the spectra of the minerals present in the data cube.

The true abundance maps of the real-world data are not available, therefore the previous works in sparse unmixing (e.g., [102], [183]) use the real-world data for qualitative analysis. We follow the same approach and provide the results for visual comparison. We use the Tricorder classification map as a reference. Note that, this reference map should not be considered as the ground truth for abundance maps. This is because, a) the Tricorder map was computed in 1995, whereas the available data set was collected in 1997 by AVIRIS, and b) the Tricorder map considers each pixel to be pure and classifies it as a member of a class that is correlated to the representative mineral in the pixel. As unmixing performs sub-pixel classification, the computed results can be different from the classification results. Nevertheless, the Tricorder map serves as a good reference for the visual comparison of the sparse unmixing solutions.

Fig. 3.11 compares the proposed algorithms with the other popular unmixing algorithms for the abundance map estimation. The figure includes only those algorithms that showed good results in practical time. The parameter settings of all the algorithms are kept same as in Section 3.6.3. From left to right, the first row of the figure shows the distribution maps of Tricorder for Alunite, Calcite,
Figure 3.10: Mineral map of Cuprite mining district Nevada: The area inside the red rectangle corresponds to the hyperspectral data cube used in the experiments. The map is made available by USGS at http://speclab.cr.usgs.gov/cuprite95.tgif.2.2um_map.gif.

Dickite and Chalcedony respectively. The second and the third rows show the corresponding abundance maps computed by OMP-Star+ and OMP-Star, respectively. It is easy to see that these algorithms have generally computed high abundances at the pixels where the distribution maps show existence of the minerals. The other algorithms have also shown similar results, however their estimated fractional abundances generally have lower values than those of the proposed algorithms. Results of the greedy algorithms in the figure were obtained in the same manner as those obtained for Tables 3.1 and 3.2.

3.8 Complexity Analysis

If $m$ is the number of spectral bands, $k$ is the number of spectra in $D$ and $p$ is the cardinality of the mixed pixel, then the per pixel computational complexity for the proposed algorithms is lower bounded by $O(mkp)$ and upper bounded by $O(Emkp)$. Here, $E$ is the product of the parameter $f$ and the cardinality of the set $T$, averaged over the algorithm iterations. The lower bound is achieved when $|T| = 1$ in all the iterations of the algorithm. In that case, the algorithm does not use the ForwardSelection method at all. Therefore, its computational complexity becomes the same as OMP, which is $O(mkp)$ [60]. The upper bound is achieved when $|T| > 1$ in all the iterations. In general, we give the computational complexity
3.8. Complexity Analysis

Figure 3.11: Qualitative comparison of fractional abundance maps: The top row shows the distribution maps of Tricorder software for the 350 × 350 pixel AVIRIS Cuprite scene shown in the red boundary in Fig. 3.10. From left to right, the maps correspond to Alunite, Calcite, Dickite and Chalcedony respectively. Each row below shows the fractional abundance maps calculated by a sparse unmixing algorithm. From top to bottom the rows correspond to OMP-Star+, OMP-Star, CSUnSAL, SUnSAL+, CSUnSAL+ and CoSaMP, respectively.
Figure 3.12: Example of typical iterations of OMP-Star: The algorithm operates on a data cube of 30 mixed pixels with known $p = 5$. For each pixel the algorithm performs 5 iterations. The five bars for each pixel show the cardinality of $\mathcal{T}$ for each iteration. For $t = 0.92$, $|\mathcal{T}| = 1$ for 63 out of total 150 iterations.

Table 3.3: Processing time (in seconds) for unmixing 500 pixel data cube with 35 dB SNR using white noise. Each pixel is a mixture of 5 randomly selected materials. Time is computed on a desktop PC equipped with an Intel Core i7-2600 CPU (at 3.4 GHz) and 8 GB RAM.

<table>
<thead>
<tr>
<th>Algo.</th>
<th>A*OMP</th>
<th>CSUnSAL+</th>
<th>SUnSAL+</th>
<th>CSUnSAL</th>
<th>LAOMP</th>
<th>SUnSAL</th>
<th>OMP-Star+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (s)</td>
<td>3812.8</td>
<td>49.9</td>
<td>48.8</td>
<td>33.5</td>
<td>32.5</td>
<td>31.4</td>
<td>19.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algo.</th>
<th>OMP-Star</th>
<th>CoSaMP</th>
<th>OMP+</th>
<th>OMP</th>
<th>ROMP</th>
<th>gOMP</th>
<th>SP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (s)</td>
<td>14.4</td>
<td>3.9</td>
<td>3.1</td>
<td>2.8</td>
<td>1.9</td>
<td>1.7</td>
<td>1.6</td>
</tr>
</tbody>
</table>

of the proposed algorithms as $O(emkp)$, where $1 \leq e \leq E$. In the experiments discussed in the earlier sections, $e$ was of the order of 4 for the proposed algorithms. We illustrate this calculation with the help of Fig. 3.12. The figure shows the value of $|\mathcal{T}|$ for five iterations of OMP-Star, per pixel, for a data cube of 30 mixed pixels. We chose $t = 0.92$ and the cardinality of the mixed pixels is set to 5. The figure shows that $|\mathcal{T}| = 1$ for 63 iterations out of the total 150 iterations performed by the algorithm. Each of these 63 iterations have the same computational complexity as that of an OMP iteration. For the rest 87 pixels the average value of $|\mathcal{T}|$ is 3.42. With $f = 2$, the algorithm performs a total of $87 \times 3.42 \times 2 \approx 595$ OMP-like iterations for these pixels. Thus, OMP-Star has the computational complexity of $63mkp + 595mkp = 658mkp$ for the whole data. Which gives $e = 658/150 \approx 4.4$ for the complete data cube.

The rest of the greedy algorithms used in this work show the worst case computational complexities of the same order as OMP [205], [60], except LAOMP and
3.9 Conclusion

A*OMP. LAOMP has the complexity $O(mkp^2L)$, where $L$ is the number of columns used in the identification step of the algorithm (see Section 3.3). The computational complexity of A*OMP depends upon the chosen cost function, path pruning strategy and related parameter settings. See [108] for details. The computational complexity of the convex relaxation based algorithms used in this work, has been reported $O(k^2)$ per iteration [68]. Table 3.3 compares the processing times of MATLAB implementations of all the algorithms used in this work. The timings were computed for a hyperspectral data cube of 500 pixels with SNR = 35dB using white noise and $p = 5$. Except for the proposed algorithms, we assumed prior knowledge of the value of $p$ for all the greedy algorithms. The proposed algorithms use the stopping rule (c) in line ‘17’ of Algorithm 2, with $\beta = 0.9$. $t = 0.92$ and $f = 2$. Furthermore, the proposed algorithms use the four-step strategy, mentioned in Section 3.6.4 to estimated the fractional abundances. Where required, we use the \texttt{lsqnonneg} procedure of MATLAB to implement the non-negative least squares method. All the algorithms utilize the same parameter settings as discussed in the previous sections.

3.9 Conclusion

This work proposes a greedy pursuit algorithm, called OMP-Star, for sparse unmixing of hyperspectral data. OMP-Star is a pixel-based algorithm that uses a futuristic greedy approach. This approach is inspired by a popular search algorithm, called A-Star. OMP-Star shows robustness against the high coherence of the data. This work also enhances OMP-Star to its non-negative variant, called OMP-Star+. This constrained version of the algorithm exploits the fact that the fractional abundances of endmembers are non-negative quantities. We propose to preprocess the hyperspectral data for greedy algorithms by taking its derivative. Generally, the derivative operation reduces the correlation among the spectral signatures, thereby improving the accuracy of sparse approximation results. However, this operation is sensitive to noise. Therefore, we explicitly evaluate derivatives for sparse unmixing and devise a strategy to use them with greedy algorithms. We test the proposed approach thoroughly on simulated as well as real-world hyperspectral data. The results demonstrate high effectiveness of the proposed approach.
CHAPTER 4

SUnGP: A Greedy Sparse Approximation Algorithm for Hyperspectral Unmixing

Abstract

Spectra measured at a pixel of a remote sensing hyperspectral sensor is usually a mixture of multiple spectra (endmembers) of different materials on the ground. Hyperspectral unmixing aims at identifying the endmembers and their proportions (fractional abundances) in the mixed pixels. Hyperspectral unmixing has recently been casted into a sparse approximation problem and greedy sparse approximation approaches are considered desirable for solving it. However, the high correlation among the spectra of different materials seriously affects the accuracy of the greedy algorithms. We propose a greedy sparse approximation algorithm, called SUnGP, for unmixing of hyperspectral data. SUnGP shows high robustness against the correlation of the spectra of materials. The algorithm employs a subspace pruning strategy for the identification of the endmembers. Experiments show that the proposed algorithm not only outperforms the state of the art greedy algorithms, its accuracy is comparable to the algorithms based on the convex relaxation of the problem, but with a considerable computational advantage.

Keywords: Hyperspectral unmixing, sparse coding, greedy pursuit.

4.1 Introduction

Modern spaceborne and airborne hyperspectral sensors measure the reflectance of the Earth’s surface at hundreds of contiguous narrow bands [189], which results in hyperspectral image cubes with two spatial and one spectral dimension (see Fig. 4.1). Each pixel of a hyperspectral image is a vector that represents a spectral signature measured by the sensor. Due to low spatial resolution of the sensors and multiple scatterings, the spectra at a pixel is usually a mixture of multiple pure spectra (endmembers), corresponding to different materials on the ground. Hyperspectral unmixing aims at identifying the endmembers in a mixed pixel and computing their fractional abundances (i.e. their proportion in the pixel) [25]. Unmixing of the hyperspectral images is considered a major challenge in remote sensing data analysis [24].

*Published in Proc. of International Conference on Pattern Recognition, 2014.*
4.1. Introduction

Figure 4.1: Illustration of a hyperspectral image cube: The XY-plane shows the spatial dimensions and the wavelength axis shows the spectral dimension. Pixels are recorded as vectors of reflectance values at different wavelengths. The cube shows reflectance patterns at ten wavelength bands. The data is collected by NASA’s AVIRIS [80] over the Cuprite mines, NV.

Recently, Linear Mixing Model (LMM) has gained considerable attention for hyperspectral unmixing [25]. This model assumes that a mixed pixel is a linear combination of its constituent endmembers, weighted by their fractional abundances. Works that employ LMM, often use the geometric properties of hyperspectral data in unmixing. They exploit the fact that the convex hull of the pure endmembers in the data forms a simplex. Thus, finding the endmembers simplifies to finding the vertices of a simplex. Classical geometric approaches assume the presence of pure pixels for each endmember in the image. Vertex Component Analysis [146], N-FINDER [211], Pixel Purity Index [28] and Simplex Growing Algorithm [43] are some of the popular examples of such approaches.

In real world data, pure pixels are not usually present for each endmember in the image. Therefore, some approaches focus on extracting the pure spectral signatures from the image for hyperspectral unmixing. Minimum Volume Simplex Analysis [123], Iterative Constrained Endmembers (ICE) [17] and Sparsity Promoting ICE [230] are examples of such approaches. Extraction of pure spectra from the images fails in highly mixed scenarios. In this case, the above mentioned approaches generate artificial endmembers that cannot be associated with true materials [101]. For highly mixed scenarios, hyperspectral unmixing is often formulated as a statistical inferencing problem under the Bayesian paradigm [24]. However, the statistical inferencing process of the Bayesian approach is generally computationally expensive.

To overcome the above issues, unmixing has recently been formulated as a sparse approximation problem [102]. This approach assumes that a mixed pixel can be approximated by a sparse linear combination of pure spectra, already available in a dictionary. Iordache et al. [102] have analyzed different sparse approximation algo-
rithms for hyperspectral unmixing, including the greedy algorithms (e.g., Orthogonal Matching Pursuit (OMP) [33]). Generally, the greedy algorithms are computationally more efficient than their convex relaxation based counterparts [31] and admit to simpler and faster implementations [196]. However, [102] shows that in comparison to the convex relaxation based algorithms (e.g., Basis Pursuit [48]), the accuracy of the greedy algorithms is severely affected by the high correlation of the spectra.

Shi et al. [183] argue strongly to exploit the greedy sparse approximation technique for hyperspectral unmixing because of its computational advantages. The authors have proposed a greedy algorithm for hyperspectral unmixing, called Simultaneous Matching Pursuit (SMP). SMP mitigates the problems caused by the high correlation of the spectra by processing the images in terms of spatial patches. The contextual information in a patch is exploited in identifying the endmembers. However, the image patch size becomes an important parameter for SMP and this parameter is image dependent. Furthermore, SMP assumes the existence of the contextual information in the complete image, which compromises its performance in the highly mixed scenarios, where this information is scarce.

In this work, we propose a greedy sparse approximation algorithm for hyperspectral unmixing. The algorithm performs Sparse Unmixing via the Greedy Pursuit strategy [33], hence it is named SUnGP. SUnGP approximates the mixed pixel by iteratively identifying its endmembers from a fixed dictionary. In each iteration, it selects a subspace that is further pruned to identify the endmember. The pruning strategy helps SUnGP in mitigating the adverse effects of the high correlation of the spectra. SUnGP is a pixel-based greedy algorithm, therefore its performance does not degrade in the highly mixed scenarios. Additionally, SUnGP can be modified to take advantage of the contextual information in the image, following the guidelines in [199]. Experiments with synthetic and real world hyperspectral data show that the proposed algorithm outperforms the existing state of the art greedy algorithms. Moreover, its results are comparable to the convex relaxation based sparse approximation algorithms, with a considerable computational advantage. In this work, we preprocess the hyperspectral data using spectral derivatives [19]. This results in better performance of greedy pursuit algorithms in hyperspectral unmixing.

4.2 Unmixing as Sparse Approximation Problem

4.2.1 Linear Mixing Model

Sparse approximation can exploit the Linear Mixing Model (LMM) of the pixels in hyperspectral unmixing. This model assumes that the reflectance $y_i$, measured at
4.2. Unmixing as Sparse Approximation Problem

the $i^{th}$ band of a mixed pixel, is a linear combination of the endmember reflectances at that band. Mathematically,
\[ y_i = \sum_{j=1}^{p} l_{ij} \alpha_j + \epsilon_i, \] (4.1)
where $p$ is the total number of the endmembers in the pixel, $l_{ij}$ is the reflectance of the $j^{th}$ endmember at band $i$, $\alpha_j$ is the fractional abundance of the $j^{th}$ endmember and $\epsilon_i$ represents the noise. If the image is acquired by a sensor with $m$ spectral channels, we can write the LMM in the matrix form:
\[ y = L\alpha + \epsilon, \] (4.2)
where, $y \in \mathbb{R}^m$ represents the reflectance vector of the pixel, $L \in \mathbb{R}^{m \times p}$ is a matrix with $p$ endmembers, $\alpha \in \mathbb{R}^p$ contains the corresponding fractional abundances of the endmembers and $\epsilon \in \mathbb{R}^m$ represents the noise. The fractional abundances follow two constraints in LMM [24]. (1) ANC: Abundance Non-negativity Constraint ($\forall_i, i \in \{1, ..., p\}, \alpha_i > 0$) and (2) ASC: Abundance Sum-to-one Constraint ($\sum_{i=1}^{p} \alpha_i = 1$). These constraints represent the fact that the fractional abundances are non-negative quantities that sum to one in a mixed pixel.

4.2.2 Sparse Approximation of a Mixed Pixel

Let $D \in \mathbb{R}^{m \times k}$ ($k > m$) be a matrix with each column $d_i \in \mathbb{R}^m$ representing the spectral signature of a material. According to LMM, if $D$ contains a large collection of spectra, including the endmembers of the mixed pixel, then:
\[ y \approx D\alpha, \] (4.3)
where $\alpha \in \mathbb{R}^k$ has only $p$ non-zero coefficients. Generally, in the remote sensing hyperspectral images, a pixel is a mixture of four to five spectra [102]. Therefore, it is safe to assume that $\alpha$ is sparse ($p \ll k$). This fact allows us to formulate hyperspectral unmixing as the following sparse approximation problem:
\[ (P^{\eta}_{0}) : \min_{\alpha} \|\alpha\|_0 \text{ s.t. } ||D\alpha - y||_2 \leq \eta, \] (4.4)
where $\|\cdot\|_0$ is the $l_0$ pseudo-norm that simply counts the number of non-zero elements in $\alpha$, and $\eta$ is the tolerance due to noise. In the sparse approximation literature, $D$ is known as the dictionary and its columns are called the atoms.

Minimization of $l_0$ pseudo-norm in $(P^{\eta}_{0})$ is, in general, an NP-hard problem [148]. In practice, its polynomial time approximation is achieved with the greedy algorithms. Relaxed convexification of the problem $(P^{\eta}_{0})$ is also possible. It is done by
replacing $l_0$ pseudo-norm in (4.4) with the $l_1$ norm of $\alpha$. Let us denote this version of the problem as $(P_1^\eta)$. Solving $(P_1^\eta)$ is equivalent to solving the well known LASSO problem [194] with an appropriate Langrangian multiplier $\lambda$ [22]. The LASSO formulation of the problem is given below as $(P_1^\lambda)$:

$$(P_1^\lambda): \min_{\alpha} \frac{1}{2}||y - D\alpha||_2 + \lambda||\alpha||_1. \quad (4.5)$$

Previous works in sparse unmixing mainly focus on solving $(P_1^\eta)$ and $(P_1^\lambda)$ [183]. For instance, Sparse Unmixing by Splitting and Augmented Langrangian (SUnSAL) [22] solves $(P_1^\lambda)$ for sparse unmixing. The authors have also enhanced this algorithm to its constrained version, called CSUnSAL. CSUnSAL solves $(P_1^\eta)$ with ASC as an additional constraint. Iordache et al. [102] have also used SUnSAL+ and CSUnSAL+ for sparse unmixing. These algorithms solve the corresponding problems by further imposing ANC on the solution space.

### 4.3 Greedy Sparse Approximation

Greedy algorithms approximate the signal $y$ in $(P_1^\eta)$ by iteratively selecting the atoms of the dictionary. These atoms are selected such that the signal is approximated in minimum number of iterations. This greedy heuristic finally results in a sparse solution. We can unify the greedy sparse approximation algorithms under a base-line algorithm, which can be stated as the following sequential steps: 1) **Identification** of the atom(s) of $D$, best correlated to the residual vector of the current approximation of $y$. For initialization, $y$ itself becomes the residual vector. 2) **Augmentation** of a selected subspace with the identified atom(s). The selected subspace is empty in the first iteration. 3) **Residual update**, after approximating $y$ with the selected subspace. The above steps are repeated until some **stopping rule** is satisfied by the algorithm. Recently proposed greedy algorithms vary in different steps of the base-line algorithms.

OMP [33] follows the base-line algorithm very closely. After identifying a single atom and augmenting the selected subspace with it, OMP finds the least squares approximation of $y$ with the selected subspace. This approximation is subtracted from $y$ in the **residual update** step. This makes the residual vector **orthogonal** to the selected subspace. This notion is an enhancement over the Matching Pursuit (MP) [138], that updates the residual vector by deflating it with the last atom added to the selected subspace. A non-negative variant of OMP, henceforth denoted as OMP+, has been proposed in [31]. OMP+ restricts its solution space only to the vectors with non-negative coefficients. Generalized-OMP (gOMP) [205] is a
generalization of OMP. It identifies a fixed number of atoms in each iteration and augments the selected subspace with all of them. Then it updates the residue similar to the OMP algorithm.

Subspace Pursuit (SP) [60], Compressive Sampling MP (CoSaMP) [150] and Regularized-OMP (ROMP) [149] are the algorithms that assume prior knowledge of the cardinality \( p \) of the signal \( y \). These algorithms identify a subspace of atoms in the step (1). In each iteration, SP identifies \( p \) atoms and augments the selected subspace with all of them. From this subspace it again selects \( p \) atoms that have maximal contribution to the approximation of \( y \) in the least squares sense. SP updates the residue like OMP. CoSaMP identifies \( 2p \) atoms in each iteration and augments the selected subspace with them. Then, it selects \( p \) atoms like SP. However, it updates the residue by using the already computed coefficients of the selected \( p \) atoms. ROMP also identifies \( p \) atoms in each iteration. Then it drops off some of these atoms using a predefined regularization rule before the step (2). In ROMP, the residual vector is also updated following OMP.

Compressive Sampling MP (CoSaMP) [150] is a greedy algorithm that works similar to gOMP in step (1) and (2) of the base-line algorithm. identifies a subspace of atoms in the step (1) of the base-line algorithm. It uses all of them to augment the selected subspace in the step (2). However, before the step (3), it drops off a fixed number of atoms from the augmented subspace that have the smallest contribution to the least squares approximation of \( y \). Regularized-OMP (ROMP) [149] also uses the same strategy; however, it uses a pre-defined regularization rule, instead of the least squares approach, to drop off the atoms. Subspace Pursuit (SP) [60] is an algorithm that assumes a prior knowledge of the cardinality \( p \) of the signal \( y \). It identifies \( p \) atoms of the library and augments the selected subspace with all of them. It then drops off the \( p \) atoms from the augmented subspace. The remaining \( p \) atoms are those that maximally contribute to the current approximation of \( y \) in the least squares sense.

4.4 Proposed Algorithm

Hyperspectral data has its own characteristics, such as, the cardinality of a mixed pixel in an image is usually small (four to five) but unknown, the spectra of different materials (i.e. the atoms of the library) are highly correlated and the fractional abundances are non-negative quantities. The greedy sparse approximation algorithms reviewed in Section 4.3 were not originally proposed for hyperspectral unmixing, therefore they do not explicitly take care of the above mentioned character-
Algorithm 4 SUnGP

Initialization:
1: Iteration: \( i = 0 \)
2: Initial solution: \( \alpha^0 = 0 \)
3: Initial residual: \( r^0 = y - D\alpha^0 = y \)
4: Selected support: \( S^0 = \text{support}\{\alpha^0\} = \emptyset \)

Main Iteration: Update iteration: \( i = i + 1 \)

Identification:
5: \( N = \{ \text{indices of the atoms of } D \text{ corresponding to the } L \text{ largest } p_j \} \)
6: \( S_t = S_{t-1} \cup N \)
7: \( \alpha_t = \min_{\alpha} \|D\alpha - y\|_2^2 \text{ s.t. } \text{support}\{\alpha\} = S_t, \alpha \geq 0 \)
8: \( j^* = \text{index } j \text{ of the largest coefficient of } \alpha_t, \text{ s.t. } j \in N \)

Augmentation:
9: \( S^t = S^{t-1} \cup \{j^*\} \)

Residual update:
10: \( r^t = y - D\alpha^t \)

Stopping rule:
11: If a) \( i > \text{desired iterations} \), or b) \( \|r^i\|_2 < \epsilon_0 \),
or c) \( \|r^i\|_2 > \beta\|r^{i-1}\|_2 \) stop, otherwise iterate again.

istics of the hyperspectral data. In fact, to the best of our knowledge, no pixel-based greedy sparse approximation algorithm has ever been proposed specifically for the problem of sparse unmixing of hyperspectral data.

Here, we present Sparse Unmixing via Greedy Pursuit (SUnGP), a pixel-based greedy algorithm that has been designed particularly for the problem of hyperspectral unmixing. SUnGP is shown in Algorithm 4. Each iteration of the algorithm comprises the three steps of the base-line algorithm in Section 4.3. In the identification step, SUnGP first computes the correlations between the atoms of the dictionary and the residual vector of the current approximation of \( y \), where \( y \) itself is considered as the residual vector at initialization. Then, SUnGP identifies the atoms of the dictionary corresponding to the \( L \) (an algorithm parameter) largest values of the computed correlations. These atoms are used for temporarily augmenting the selected subspace. Using this subspace, a non-negative least squares approximation of the mixed signal is computed (line ‘8’ in Algorithm 4). SUnGP then identifies the atom from the aforementioned \( L \) atoms that has the maximum
contribution in this approximation. This atom is used for permanently augmenting the selected subspace (line ‘10’).

Notice that, SUnGP first identifies a subspace of \( L \) atoms (which are highly correlated) and later prunes it considering the non-negativity of the solution space, to identify the single best atom. Once the best atom is identified, it is added to the selected subspace, never to be dropped off in the future iterations. This strategy stems directly from the aforementioned characteristics of the hyperspectral data. SUnGP follows OMP in the residual update step and uses a disjunction of three stopping rules (line ‘13’). The rules (a) and (b) are self explanatory. The rule (c) ensures that the algorithm stops if it is not able to reduce the \( l_2 \) norm of the residual vector at least by a fraction \( \beta \) in its last iteration. The rules (b) and (c) allow SUnGP to operate without prior information about the cardinality of the mixed pixel.

4.5 Processing the Data for Greedy Algorithms

In the sparse approximation, the correlation among the atoms of the dictionary is quantified by the mutual coherence \( \mu \in [0, 1] \) of the dictionary, as follows:

\[
\mu = \max_{i,j:j \neq i} \frac{\text{abs}(d_i^T d_j)}{||d_i||_2 ||d_j||_2},
\]

where \( d_z \) is the \( z^{th} \) atom of the dictionary. In general, the greedy sparse approximation algorithms are able to identify the support of the solution more accurately, if \( \mu \) is small for the dictionary. For instance, according to [196], if \( \mu < 0.33 \) OMP will always find the exact support for a signal that is a linear combination of two distinct atoms of the dictionary. However, in the unmixing problem, usually \( \mu \approx 1 \) [102]. Researchers at the German Aerospace Center [19] have shown that for a dictionary of spectra sampled at a constant wavelength interval, \( \mu \) can be reduced by taking the derivatives of the spectra. The derivative of a spectra \( d \in \mathbb{R}^m \) is defined as

\[
\Delta(d) = \frac{d(b_i) - d(b_j)}{b_j - b_i}, \quad \forall i \in \{1, ..., m - c\},
\]

where \( b_z \) is the wavelength at the \( z^{th} \) band, \( d(b_z) \) is the reflectance value at that wavelength and \( j = i + c \), with \( i, j \) and \( c \) being positive integers.

Keeping in view its coherence reduction ability, we benefit from the derivative operation in hyperspectral unmixing with the greedy algorithms. We propose to use the following strategy for processing the hyperspectral data for the greedy sparse approximation algorithms:
1. Create the dictionary $D_\Delta$ and the pixel $y_\Delta$ by taking the derivative of each spectra in $D$ and the pixel $y$, respectively.

2. Compute $\alpha$ with the greedy algorithm using $D_\Delta$ and $y_\Delta$.

3. Estimate the fractional abundance vector $\tilde{\alpha}$, by the non-negative least squares approximation of $y$ using the atoms of $D$ corresponding to the support of $\alpha$.

The unmixing performed on the differentiated data in step (2) is used to identify the correct support of the solution. Once the support is found, it is used for the actual estimation of the fractional abundances using the original data in the step (3). In the above strategy, we use the dictionaries after normalizing their atoms in $l_1$ norm. By doing so, the estimated fractional abundances automatically satisfy ASC. It is worth mentioning here that the derivative operation helps in coherence reduction, however the correlation among the spectra of the differentiated data still remains high enough to cause problems for the greedy sparse approximation algorithms. Therefore, the algorithms need to show robustness against the correlation of the spectra for effective hyperspectral unmixing.

4.6 Experiments

In this section, we present the results of the experiments performed with synthetic data and the real-world data. The experiments with the synthetic data are important because they provide quantitative evaluation of the approach, which is not possible with the real-world data. In all the experiments we used a fixed dictionary, which was created from the NASA Jet Propulsion Laboratory’s Advanced Space-borne Thermal Emission and Reflectance Radiometer (ASTER) spectral library (http://speclib.jpl.nasa.gov). This library contains pure spectra of 2400 materials. To perform the experiments, we selected 425 of these spectra and resampled them in the wavelength range 0.4 to 2.5 $\mu$m, at a constant interval of 10 $nm$. The resampling is performed to match the sampling strategy of the NASA’s Airborne

---

1The approach in this chapter, i.e. SUnGP, was developed in parallel to the OMP-Star algorithm developed in Chapter 3. Therefore, the results do not include comparison with OMP-Star. However, the evaluation has been done using the same metrics and protocols used in Chapter 3 for the synthetic data. Moreover, the exact same hyperspectral cube is used for the evaluation of both approaches on the real data. From these experiments, it is clear that SUnGP has a comparable performance with OMP-Star on the used data sets. Whereas OMP-Star is particularly designed to show more robustness against high correlation among the spectra, SUnGP trades off this robustness with computational efficiency. The later is more than twice as fast as the former.
4.6. Experiments

Visible and Infrared Imaging Spectrometer (AVIRIS) [80]. We dropped 24 bands of the spectra in the dictionary because of zero or very low reflectance values. This made $D$ a $200 \times 425$ matrix. The spectra were selected such that $\mu = 0.9986$ for the dictionary. We kept $\mu < 1$ in order to ensure that the spectra in the dictionary are unique.

4.6.1 Results on Synthetic Data

We simulated the synthetic hyperspectral data with 500 mixed pixels, where each pixel was a linear combination of $p$ randomly selected spectra from the dictionary. Following the experimental protocol in [102], we drew the fractional abundances of the endmembers in each pixel from a Dirichlet distribution. Therefore, the fractional abundances satisfy ASC. We added the Gaussian white noise to the data such that the pixels had $\text{SNR} = 50 \text{dB}$. The algorithms were evaluated for the two goals of hyperspectral unmixing, namely, the endmembers identification and the estimation of the fractional abundances. For the former, we used the evaluation metric of unmixing fidelity $\Phi(\alpha) \rightarrow [0, 1]$. If $P = \{x \mid x$ is the index of an endmember in $D\}$ and $A = \{a \mid a$ is a non-zero elements in $\alpha\}$, then:

$$\Phi(\alpha) = \frac{|P \cap A|}{|A|},$$

(4.8)

where $|$ denotes the cardinality of the set. Fig. 4.2a shows the results of the experiments performed to evaluate the endmember identification ability of the greedy algorithms. The values in these results, and the results to follow, are the mean values of the metrics, computed over the whole synthetic data. The results show that SUnGP performs better than the existing greedy sparse approximation algorithms. For SUnGP we used $L = 50$, which was optimized on a separate training data set. Different parameters of the other algorithms were also optimized on the same training data set. We have given the correct value of $p$ to the greedy algorithms as their input parameters. gOMP is tuned to select 2 atoms in its each iteration. The figure shows some of the results in dotted plots. The algorithms corresponding to these plots assume a priori knowledge of $p$. Therefore, they are not of practical use in hyperspectral unmixing. However, we have included them in the analysis for a comprehensive comparison of SUnGP’s performance with the state of the art greedy algorithms. Fig. 4.2b compares the algorithms for the different noise levels of data. SUnGP again shows good performance, especially at high $\text{SNR}^2$.

\footnote{SNR as high as 400dB is a realistic value for hyperspectral remote sensing instruments [19]. Our analysis focuses only on the more challenging case of low SNR.}
Table 4.1: Processing time (seconds) for 500 pixel synthetic image with 50 dB SNR. Each pixel is a mixture of 5 randomly selected spectra. The time is computed on a desktop PC equipped with an Intel processor at 3.4 GHz and 8 GB RAM.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (s)</td>
<td>0.83</td>
<td>0.85</td>
<td>0.89</td>
<td>1.76</td>
<td>2.27</td>
<td>2.82</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algo.</th>
<th>SUnGP</th>
<th>SUnSAL[22]</th>
<th>CSUnSAL[22]</th>
<th>SUnSAL+[22]</th>
<th>CSUnSAL+[22]</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (s)</td>
<td>5.11</td>
<td>31.3</td>
<td>33.7</td>
<td>35.9</td>
<td>36.1</td>
<td>-</td>
</tr>
</tbody>
</table>

In our experiments, we have processed the data according to the strategy discussed in Section 4.5. The unmixing fidelity of the solution can be computed at the step (2) of the strategy. Therefore, the results discussed above were computed at that step. Following [6], we chose $c = 2$ in Equation (4.7) for the derivative operation. To evaluate the estimation of the fractional abundances we also performed the step (3) of the strategy and compared the results of SUnGP with the results of the popular convex relaxation based algorithms that have been proposed specifically for hyperspectral unmixing. For this comparison, we did not assume a priori knowledge of $p$ and used the stopping rule (c) (line ‘13’, Algorithm 4) with $\beta = 0.9$ for SUnGP. The value of $\beta$ was optimized on a separate training data set. For the convex relaxation based algorithms we chose $\lambda = 10^{-3}$. This value was optimized on the training data. We used the Euclidean distance between the estimated fractional abundance vector $\tilde{\alpha}$ and the actual fractional abundance vector $\alpha_0$, as the evaluation metric. The comparison of the results is given in Fig. 4.2c, which shows that SUnGP’s performance is comparable to the convex relaxation based algorithms.

A major advantage of using the greedy algorithms in hyperspectral unmixing is the computational efficiency. Table 4.1 compares the computational timings of the algorithms. These timings have been computed for unmixing of a 500 pixel synthetic image, with $p = 5$ for each pixel and SNR = 50dB. Greedy algorithms, other than SUnGP, assume prior knowledge of $p$. The algorithms use the same parameter settings which are used in the results discussed above. Timings have been computed with authors’ MATLAB implementations for each of the algorithms.

4.6.2 Results on the Real-world Data

We also performed sparse unmixing of the real-world hyperspectral data (http://aviris.jpl.nasa.gov/data/free_data.html), acquired by AVIRIS. From this
4.6. Experiments

Figure 4.2: Comparison of the results on synthetic data: a) Unmixing fidelity computed by different algorithms, as a function of the cardinality of the mixed pixels. The image contains the Gaussian white noise with SNR = 50dB for each pixel. b) Unmixing fidelity as a function of the SNR of the pixels. The cardinality of each pixel is 5. c) Euclidean distance between $\alpha_0$ and $\tilde{\alpha}$, as a function of the pixel cardinality. The comparison is between SUnGP and the convex relaxation algorithms.

For the region analyzed in this work, we separately show the classification map of Alunite (a mineral) computed by Tricorder. The figure also shows the fractional abundance maps of Alunite as computed by the different sparse approximation algorithms (only the best ones are shown because of the space limitations). From the figure, it is visible that SUnGP has estimated high fractional abundances for the pixels which have been classified as Alunite by Tricorder. The values are higher than those computed by any other algorithm.
Figure 4.3: Comparison of the results on the real-world data: The USGS classification map shows the labels of different materials in an image taken over Cuprite mines, NV. The labels are assigned with the Tricorder software. The 350 × 350 area inside the rectangle is analyzed in this work. The Tricorder map shows the classification map of Alunite (a mineral). Each of the other four maps show the fractional abundances of Alunite, computed by the algorithms mentioned.

Following [102] and [183], we have provided the results only for the visual comparison. The quantitative comparison of the results on the real-world data is not possible, as no ground truth values of the fractional abundances are available for the real-world data [183]. In the results shown, we have used 188 spectral bands (out of 224) of the image cube. The other bands were dropped because of zero or very low reflectance values. The corresponding bands were also dropped from the dictionary. We used the strategy discussed in Section 4.5 for the greedy algorithms. For CoSaMP we set $p = 5$. Rest of the algorithms use the same parameter settings as discussed in the previous section.

4.7 Conclusion

We have proposed a pixel based greedy sparse approximation algorithm, called SUnGP, for hyperspectral unmixing. The proposed algorithm identifies different spectra in a mixed hyperspectral pixel by iteratively selecting a subspace of spectra from a fixed dictionary and pruning it. SUnGP has been tested on synthetic as well as the real-world remote sensing hyperspectral data. The algorithm has been shown to outperform the existing state of the art greedy sparse approximation algorithms. Furthermore, its results are comparable to the convex relaxation based sparse approximation algorithms, with a considerable computational advantage.
CHAPTER 5

RCMF: Robust Constrained Matrix Factorization for Hyperspectral Unmixing

Abstract

We propose a constrained matrix factorization approach for linear unmixing of hyperspectral data. Our approach factorizes a hyperspectral cube into its constituent endmembers and their fractional abundances, such that the endmembers are sparse non-negative linear combinations of the observed spectra themselves. The association between the extracted endmembers and the observed spectra is explicitly noted for physical interpretability. To ensure reliable unmixing, we make the matrix factorization procedure robust to outliers in the observed spectra. Our approach simultaneously computes the endmembers and their abundances in an efficient and unsupervised manner. The extracted endmembers are non-negative quantities, whereas their abundances additionally follow the sum-to-one constraint. We thoroughly evaluate our approach using synthetic data with white and correlated noise, as well as real hyperspectral data. Experimental results establish the effectiveness of our approach.

Keywords: Hyperspectral unmixing, robust matrix factorization, sparse representation, blind source separation, unsupervised unmixing.

5.1 Introduction

Hyperspectral imaging acquires precise spectral information of the scene radiance that is exploited for efficient Earth exploration in remote sensing. Nevertheless, contemporary hyperspectral imaging lacks in spatial resolution [5], [120], causing a pixel of a remotely sensed image to generally correspond to a large area on the ground (see Fig. 5.1). This causes the spectra sensed at a pixel to be a mixture of reflectances of different materials present in that area. Moreover, multiple scatterings of light and the presence of intimate material mixtures on the ground also result in mixing of the sensed material spectra [24]. Identifying materials on the Earth’s surface by extracting their pure spectral signatures (endmembers) and computing their proportions (fractional abundances) in a hyperspectral pixel are the two fundamental tasks handled by hyperspectral unmixing.
Chapter 5. RCMF for Hyperspectral Unmixing

Figure 5.1: Hyperspectral images comprise hundreds of spectral channels (fewer channels are shown for illustration), but a pixel usually corresponds to a large area on the ground, making it a mixture of reflectance spectra of multiple materials.

To unmix a pixel, it is common to model it as a linear combination of its constituent endmembers [25], [81]. Such a modeling is effective when materials occur in spatially distinct regions on the Earth’s surface and minimal light scattering is observed in the scene [24]. This work also focuses on the Linear Mixing Model (LMM) of the spectra [110]. Under LMM, the convex geometry of the observed spectra is often exploited by the unmixing approaches [27], [41], [57], [160]. These approaches identify the endmembers as the vertices of a simplex, formed by the convex hull of the endmembers. These techniques assume the presence of at least one pure endmember pixel in the image. The Simplex Growing Algorithm [43], Vector Component Analysis [146], N-FINDR [211], Pixel Purity Index [28], Iterative Error Analysis [151] and Successive Volume Maximization [42] are some classic examples in this direction.

In practice, pure pixels are not always present [102], [4] for each endmember in a hyperspectral image. Hence, approaches like Iterative Constrained Endmembers (ICE) [17], Minimum Volume Simplex Analysis [123] and Sparsity promoting ICE [230] tend to generate the endmembers from the image itself. Nevertheless, these approaches are computationally expensive [21] and they do not perform well in highly mixed scenarios [102]. Formulating hyperspectral unmixing as a statistical inference problem under the Bayesian framework improves performance for such scenarios [24]. However, the computational complexity of Bayesian methods generally remains prohibitive [6].
5.1. Introduction

More recently, Iordache et al. [102] has shown the effectiveness of sparse regression methods [48], [33], [22] for hyperspectral unmixing by approaching the problem in a supervised manner. Their approach assumes prior knowledge of the potential endmembers in an image, and represents a mixed pixel as a sparse linear combination of those endmembers. The success of this framework has led to numerous efforts in tailoring sparse regression algorithms for sparse unmixing, e.g. [6], [183]-[217]. Although useful, the sparse unmixing framework relies on the exhaustiveness of the dictionary comprising the potential endmembers in the image. Whereas larger dictionaries are required to ensure correct identification of the endmembers, increasing the dictionary size generally results in increasing its coherence [196]. It is well known that high coherence of the dictionary can lead to poor performance of the sparse regression framework [70].

Hyperspectral unmixing can be readily formulated as a Non-negative Matrix Factorization (NMF) problem [161], widely solved for blind source separation [107]. A major advantage of NMF over sparse unmixing [102] is that it does not assume the potential endmembers to be known a priori. Nevertheless, the non-convexity of the problem usually results in the solution that are only locally optimal. In general, this issue is resolved by adding more constraints to the problem, giving rise to a variety of constrained NMF approaches, e.g. [83]-[169]. However, most of these approaches underperform for hyperspectral unmixing, as they are not originally proposed for this purpose [228]. Pauca et al. [158] first imposed a spectral smoothness constraint over NMF for spectral data analysis. However, as in sparse unmixing, their approach requires a priori known library of endmembers for effective unmixing. Miao and Qi [143] used a minimum volume constraint with NMF for extracting the endmember. Nevertheless, their technique necessarily requires a data dimensionality reduction step for unmixing, which can result in the loss of useful information.

Jian and Qian [103] incorporated a piecewise smoothness constraint into NMF for spectral unmixing. Lu et al. [130] proposed a manifold learning and sparse NMF based hyperspectral unmixing approach that considers local space information for improved performance. Later, Lu et al. [129] also proposed a structure constrained sparse NMF method. Qian et al. [164] constrained NMF with an $L_{1/2}$-sparsity constraint for hyperspectral unmixing. Similarly, Yuan et al. [228] introduced a substance dependence constraint into NMF to make the factorization procedure more stable. The aforementioned matrix factorization based approaches generally unmix hyperspectral data well. Nevertheless, they suffer form a common problem that they can also result in artificial endmember spectral signatures that do not
associate with real materials in the scene. These spectra are formed as by-products of the unmixing approaches themselves, but the approaches fail to provide any useful physical interpretation of these spectra.

In this work, we propose a robust constrained matrix factorization approach for hyperspectral unmixing that mitigates the problem of artificial endmember spectra. To keep the association between the extracted endmember spectra and the materials in the scene, our approach constrains the spectral signatures to be linear combinations of the image pixels themselves. In this manner, presence of any pure material pixel in the image is also readily exploited by our approach. Similarly, it is also able to effectively utilize the pixels with dominant endmembers because extracting the endmember spectra from these pixels is easier than to compute them as pure mathematical outcomes. Our approach explicitly notes the contributions of the pixels to the extracted endmember spectra. This helps in physical interpretability of the extracted spectra. Since the endmember spectra are constructed using the observed data, we make the approach robust to any outliers present in the data. To achieve our objectives, we reformulate the unmixing problem to incorporate the said physical interpretability of the endmembers and systematically derive an efficient optimization algorithm to solve it. The algorithm simultaneously extracts the endmember spectral signatures and computes their abundances.

In this work, we also propose a non-negative variant of the Subspace Pursuit algorithm, that is known for its computational efficiency [6]. The proposed Non-negative Subspace Pursuit is exploited in our approach for efficient matrix factorization. The endmembers computed by our approach are non-negative values and their estimated abundances additionally follow the well-known sum-to-one constraint [24]. We show the effectiveness of our approach on synthetic and real hyperspectral data.

5.2 Problem formulation

5.2.1 Linear mixing model

This work focuses on the Linear Mixing Model (LMM) [110], that represents a pixel \( y \in \mathbb{R}^m \) of a hyperspectral image as:

\[
y = \Phi \alpha + \epsilon, \tag{5.1}
\]

where \( \Phi \in \mathbb{R}^{m \times K} \) contains the endmembers as its columns, \( \alpha \in \mathbb{R}^K \) encodes their fractional abundances and \( \epsilon \in \mathbb{R}^m \) represents the error, considered as additive Gaussian noise. Under this model, the coefficients \( \alpha_i \in \{0, \ldots, K\} \) of \( \alpha \) must satisfy two constrains [24]: (1) \( \forall i, \alpha_i \geq 0 \), i.e. Abundance Non-negativity Constraint (ANC), and
5.2. Problem formulation

(2) \( \sum_{i=1}^{K} \alpha_i = 1 \), i.e. Abundance Sum-to-one Constraint (ASC). These constraints signify the fact that the proportions of the endmembers in a pixel are non-negative quantities that add up to 1.

5.2.2 Unmixing as Constrained Matrix Factorization

Let \( Y \in \mathbb{R}^{m \times n} \) be the matrix formed by arranging the \( n \) pixels of a hyperspectral image as its columns. Assuming that \( \Phi \) now contains all the endmembers in the whole image, we can compactly write LMM as follows:

\[
Y = \Phi A + E,
\]

where \( A \in \mathbb{R}^{K \times n} \) and \( E \in \mathbb{R}^{m \times n} \) are the abundance matrix and the noise matrix, respectively. In this work, both \( \Phi \) and \( A \) are considered to be unknown, making unmixing a blind source separation problem. Incorporating ANC and ASC in (5.2) results in:

\[
Y = \Phi A + E \quad \text{s.t.} \quad \forall i,j, \alpha_{i,j} \geq 0; ||\alpha_j||_1 = 1
\]

(5.3)

where, \( \alpha_{i,j} \) is the coefficient of \( A \) at index \((i,j)\), \( \alpha_j \) denotes the \( j \)th column of \( A \) and \( ||.||_1 \) computes the \( \ell_1 \)-norm.

As \( \Phi \) contains the endmembers of the complete image and \( \alpha_j \) corresponds to the \( j \)th pixel only, we can expect \( \alpha_j \) to be generally sparse. Nevertheless, an explicit sparsity constraint over this vector can be ignored because restricting the \( \ell_1 \)-norm of this vector already induces a form of sparsity in it [49]. That is, many coefficients of this non-negative vector shrink towards small values because their sum must not exceed one. This loosely emulates the sparsity induced by a Laplacian prior over the vector. We have also shown in [4] that ASC (along ANC) results in effective sparse abundance vectors for unmixing. However, if ASC is loosened/ignored in (5.3), a sparsity constraint over the abundance vector should be additionally imposed.

Since spectra are non-negative quantities, we can also force the coefficients \( \varphi_{h,i} \) of \( \Phi \) to be non-negative, resulting in:

\[
Y = \Phi A + E \quad \text{s.t.} \quad \forall i,j, \alpha_{i,j}, \varphi_{h,i} \geq 0; ||\alpha_j||_1 = 1
\]

(5.4)

Estimating \( \Phi \) and \( A \) that satisfy (5.4), is a constrained matrix factorization problem\(^1\). A matrix \( \Phi \) obtained by solving that problem, would generally approximate \( Y \)'s endmembers well. Nevertheless, some of the computed endmembers in \( \Phi \) could

\(^1\text{We intentionally use a broader term than Non-negative Matrix Factorization because the problem contains more than just non-negativity constraints.}\)
also be artificial. Artificial endmembers do not belong to any real material in the scene, but exist only in the solution space of the problem due to its inherent non-convexity, which also makes their physical interpretation hard. In order to mitigate this issue, we reformulate the model in (5.4) as follows.

We force the columns \( \varphi_{i \in \{1, \ldots, K\}} \) of \( \Phi \) to be non-negative combinations of the image pixels themselves. Moreover, we force each \( \varphi_i \) to only use at most \( k \) pixels in its construction, where \( k \) is a small positive integer. Concretely, our model becomes:

\[
Y = Y \Xi A + E \\
\text{s.t.} \forall i, j, \xi_{j,i}, \alpha_{i,j} \geq 0; \| \alpha_j \|_1 = 1; \| \xi_i \|_o \leq k, \tag{5.5}
\]

where \( \Xi \in \mathbb{R}^{n \times K} \) is a matrix with coefficients \( \xi_{j,i} \) and \( \xi_i \) as its \( i \)th column. The symbol \( \| . \|_o \) denotes the \( \ell_o \) pseudo-norm, that counts the number of non-zero coefficients in a vector. Since both \( Y \) and \( \Xi \) are non-negative, the endmembers in \( \Phi = Y \Xi \) also remain non-negative in the above formulation.

At this point, it is worth mentioning that Ambikapathi et al. [11] observed an important geometric property of hyperspectral data, that convex-hull of pixels lies strictly inside the convex hull of the endmembers. Considering this, the model in (5.5) may appear restrictive because constructing endmembers as non-negative combinations of the pixels seems improbable under this observation. However, this is not true. Note that, the observation is valid only in noise-free settings [11]. In (5.5), we always consider \( E \) to be a non-zero matrix, which also qualifies the scope of this work to the practical noisy scenarios. In noisy settings, the convex hull of the observed pixels is not necessarily bounded by the convex hull of the endmembers. Not to mention, we do not restrict \( \| \xi_i \|_1 \) to 1 in (5.5), which would be required to define the convex hull of the pixels. At practical noise levels, our model accurately reconstructs the endmembers, which is verified by the unmixing accuracy of our approach in Sections 5.4 and 5.5.

The model in (5.5) records the relationship between the endmembers and the pixels in matrix \( \Xi \). Hence, it is also able to explicate any pixels that it considers to be the pure endmembers\(^2\). For such pixels, \( \| \xi_i \|_o \) becomes exactly 1. This characteristic of the model is certainly desirable, which also justifies the use of the \( \ell_o \)-sparsity constraint. However, there is one subtle issue. Since the model requires the endmembers to be constructed from the pixels themselves, an approach for computing \( \Xi \) and \( A \) under (5.5) must ensure that the computations are robust to outliers.

\(^2\)We do not claim that such pixels would necessarily be pure endmembers, as it may only be the case that they are linearly inseparable. However, explicit identification of such pixels can be useful in further scrutiny, if desired.
5.3 Proposed Approach

We propose a constrained matrix factorization approach for hyperspectral unmixing that computes the endmembers and their abundances according to the model in Eq. (5.5). For reliable unmixing, the approach is also kept robust to any outliers among the pixels. We first describe the objective function for our approach and then explain its optimization procedure.

5.3.1 Objective function

The objective function for computing $\Xi$ and $A$ under (5.5) can be written as:

$$\min_{\Xi,A} \| Y - Y\Xi A \|_F^2 \quad \text{s.t.} \quad \forall i,j, \xi_{j,i}, \alpha_{i,j} \geq 0; \| \alpha_j \|_1 = 1; \| \xi_i \|_0 \leq k, \quad (5.6)$$

where $\| \cdot \|_F$ denotes the Frobenius norm of a matrix. In (5.6), the cost associated with the reconstruction of a pixel is quadratic. Generally, a large reconstruction error can be expected for an outlier because outliers do not follow the same distribution as the actual data. This makes the quadratic penalty for the outliers to be too strict. Hence, to render the optimization procedure less sensitive to the outliers, we force the penalty for the larger errors to become linear. This is done by modifying the objective function to the following:

$$\min_{\Xi,A,\Delta} \left\{ \| (Y - Y\Xi A)\Delta^{-\frac{1}{2}} \|_F^2 + \| \Delta \|_1 \right\} \quad \text{s.t.} \quad \forall i,j, \| \alpha_j \|_1 = 1; \xi_{j,i}, \alpha_{i,j} \geq 0; \| \xi_i \|_0 \leq k; \delta_j \geq \varepsilon, \quad (5.7)$$

where $\Delta \in \mathbb{R}^{n \times n}$ is a diagonal matrix with strictly positive diagonal entries, the $j^{th}$ of which is denoted as $\delta_j$; and $\varepsilon$ is a scalar constant.

The above modified function is an outlier robust form of (5.6). To show this, let us focus on the minimization of the cost associated with the $j^{th}$ pixel only, considering both $\Xi$ and $A$ to be fixed. In that case, (5.7) reduces to the following optimization problem:

$$\min_{\delta_j} \left\{ \| y_j - Y\Xi \alpha_j \|_2^2 + \frac{\| y_j - Y\Xi \alpha_j \|_2^2}{\delta_j} \right\} \quad \text{s.t.} \quad \delta_j \geq \varepsilon. \quad (5.8)$$

By differentiating the expression in the brackets $w.r.t.$ $\delta_j$ and equating it to zero, we can show that $h = \| y_j - Y\Xi \alpha_j \|_2$ becomes the minimizer $\delta^*_j$, which represents a linear penalty. However, this penalty becomes applicable only when $h \geq \varepsilon$. Otherwise, we must choose $\delta^*_j = \varepsilon$ because $\varepsilon$ is the closest value to the minimizer that is allowed.
by the outer constraint \( \delta_j \geq \varepsilon \). In that case, the penalty takes the quadratic form \( \left\{ \frac{\varepsilon^2}{\varepsilon} + \varepsilon \right\} \). Note that, this transformation of the penalty in our objective function is in line with the transformation of the penalty in robust linear regression [12] under the widely known Huber loss function [93].

5.3.2 Optimization algorithm

The optimization problem in (5.7) is non-convex. Nevertheless, it has the following desirable characteristics. (a) With fixed \( \Xi \) and \( \Delta \), the optimization of \( A \) becomes convex. (b) By fixing \( \Xi \) and \( A \), we get a closed-form solution for the optimal \( \Delta \). (c) With known \( A \) and \( \Delta \), \( \Xi \) can be computed by solving a constrained sparse optimization problem. The properties are further explained below. We propose Algorithm 5 that exploits these properties in employing a Block Coordinate Descent (BCD) scheme [18] to solve the optimization problem (5.7). The BCD scheme guarantees the algorithm to asymptotically converge to a stationary point in the solution space [18].

To understand the aforementioned properties of the problem and their exploitation in Algorithm 5, let us first focus on the computation of \( A \). By fixing \( \Xi \) and \( \Delta \), and replacing \( Y\Xi \) by \( \Phi \) in (5.7), we can estimate \( A \) as follows:

\[
\min_A ||Y - \Phi A||_F \quad \text{s.t. } \forall i, j, ||\alpha_j||_1 = 1; \alpha_{i,j} \geq 0.
\]

This is a Fully Constrained Least Squares (FCLS) problem that can be solved using many existing techniques, e.g. active sets method [153], Alternating Direction Method of Multipliers (ADMM) [68]. In this work, we set the implementation of SUnSAL algorithm provided by Bioucas-Dias and Figueiredo [22] to solve this problem. In Algorithm 5, this computation is carried out on line ‘2’.

To compute \( \Delta \), we separately estimate each of its diagonal entries \( \delta_j \), by solving for the objective function in (5.8). From the discussion in Section 5.3.1, it is clear that an optimal \( \delta_j^* \) can be directly computed by choosing the larger value among \( \varepsilon \) and \( ||y_j - \Phi \alpha_j||_2 \). This procedure is performed in line ‘3-6’ of the algorithm. It is worth mentioning that although the computation is element-wise, it exactly emulates solving (5.7) for the complete matrix \( \Delta \) (with fixed \( \Xi \) and \( A \)) due to the special construction of \( \Delta \), which gives us a closed-form solution of the matrix.

To compute \( \Xi \), we can write the optimization objective in (5.7) as:

\[
\min_\Xi ||(Y - Y\Xi A)\Delta^{-\frac{1}{2}}||_F^2, \text{ s.t. } \forall i, j, \xi_{i,j} \geq 0; ||\xi||_o \leq k.
\]

Notice that, we do not disregard \( \Delta \) in (5.10). This is because we intend to reformulate the objective function by changing its input argument, which will be made
Algorithm 5 Robust Constrained Matrix Factorization

**Input:** Data $Y \in \mathbb{R}^{m \times n}$ normalized in $\ell_2$-norm, number of endmembers $K$, sparsity level $k$, number of iterations $Q$.

**Initialize:** $\Xi \in \mathbb{R}^{n \times K}$ as a binary matrix with 1 appearing randomly in each column only once, $\Phi = Y\Xi$, $\Delta \in \mathbb{R}^{n \times n}$ as an identity matrix, and $q = 1$.

**Iterations:**
1. for $q = 1$ to $Q$ do
2. \[ A = \text{argmin}_A ||Y - \Phi A||_F^2 \text{ s.t. } \forall i, j, ||\alpha_{i,j}||_1 = 1; \alpha_{i,j} \geq 0 \text{ (Solve FCLS using [22])} \]
3. for $j = 1$ to $n$ do
4. \[ \delta_j^* = \max(\varepsilon, ||y_j - \Phi \alpha_j||_2) \]
5. \[ \delta_j \leftarrow \delta_j^* \]
6. end for
7. $\Gamma = Y - \Phi A$
8. for $i = 1$ to $K$ do
9. \[ \varrho = \Delta^{-\frac{1}{2}} \alpha^{T} \]
10. \[ \psi = \frac{\Gamma \varrho}{\alpha^{T} \varrho} + Y \xi_i \]
11. \[ \xi_i^* = \text{argmin}_{\xi_i^*} ||\psi - Y \xi_i^*||_2^2 \text{ s.t. } \forall j, \xi_{j,i}^* \geq 0; ||\xi_i^*||_o \leq k \text{ (Solve using Algorithm 6)} \]
12. \[ \Gamma = \Gamma + Y(\xi_i - \xi_i^*) \alpha_i \]
13. \[ \xi_i \leftarrow \xi_i^* \]
14. end for
15. $\Phi = Y\Xi$
16. end for

**Output:**
17. Endmember matrix $\Phi$, Abundance matrix $A$. 

5.3. Proposed Approach
clear shortly. In that case, rescaling of the new argument by $\Delta$ must also be taken into consideration for the correct reformulation.

Let us briefly ignore the outer constraints in (5.10). We can minimize the quadratic loss in the remaining objective by separately minimizing the cost incurred by each column of $\Xi$. This can be done by solving the following problem:

$$\forall i, \min_{\xi_i} \left\| Y \Delta^{-\frac{1}{2}} - Y (\Xi A - \xi_i \alpha^i + \xi_i^* \alpha^i) \Delta^{-\frac{1}{2}} \right\|_F^2,$$

(5.11)

where $\alpha^i \in \mathbb{R}^{1 \times n}$ denotes the $i$th row of $A$ and $\xi_i^*$ is an updated version of $\xi_i$ that would minimize the loss. By changing the input argument from $\xi_i$ to $\xi_i^*$, $\Xi$ also becomes a constant in (5.11). Exploiting the fixed matrices, we can further modify our optimization objective as

$$\forall i, \min_{\xi_i^*} \left\| \left\{ \frac{(Y - Y \Xi A) \Delta^{-\frac{1}{2}} \alpha^i \alpha^i} + Y \xi_i \right\} - Y \xi_i^* \right\|_2^2,$$

(5.12)

where the superscript ‘$\top$’ signifies the transpose operation. Let us denote the expression within the braces by $\psi$ to finally arrive at the following form of (5.10):

$$\forall i, \min_{\xi_i^*} ||\psi - Y \xi_i^*||_2^2 \text{ s.t. } \forall j, \xi_{j,i}^* \geq 0; ||\xi_i^*||_\alpha \leq k.$$

(5.13)

Above, is an objective function of a sparse optimization problem, with a non-negativity constraint over the sparse codes $\xi_i^*$. As mentioned in line ‘11’ of the algorithm, we use Algorithm 6 to solve this problem. For the sake of continuity, we momentarily defer the discussion on Algorithm 6 to the next paragraph. The computation and the update procedure for the matrix $\Xi$ are given in lines ‘7 - 14’ of Algorithm 5. In line ‘7’, we compute the residue matrix $\Gamma \in \mathbb{R}^{m \times n}$ outside the for-loop that iterates over the columns of $\Xi$. In line ‘12’, we update $\Gamma$ to account for the change in the residue caused by the newly computed $\xi_i^*$, before updating $\xi_i$ in line ‘13’ of the algorithm.

It is possible to use or extend existing algorithms [6], [33], [138], [150], [205], [60] to solve (5.13). These algorithms employ a common optimization strategy, known as the greedy pursuit in the sparse representation literature [70]. A detailed discussion on this strategy and the above referred algorithms can be found in our previous work [6]. In that work, a comprehensive analysis of these algorithms revealed that the Subspace Pursuit (SP) [60] can solve (5.13) very efficiently, however, without accounting for the non-negativity constraint. Hence, in this work, we extend SP to additionally incorporate the non-negativity constraint and use it to solve (5.13). The non-negative variant of SP has analogous computational advantages over the
5.3. Proposed Approach

non-negative variants of the existing algorithms. To the best of our knowledge, such an extension of SP has not been previously proposed. The Non-negative Subspace Pursuit (NSP) is given in Algorithm 6. We note that a non-negative variant of Orthogonal Matching Pursuit algorithm (OMP) [33] has been proposed by Brukenstein et al. [31] that can be used to solve (5.13). However, compared to that algorithm, NSP shows more robustness against the local optimality of the greedy pursuit strategy, just as SP is more robust than OMP in this regard.

Algorithm 6 Non-negative Subspace Pursuit

Input: Sparsity level \( k \), data \( \psi \), dictionary \( Y \).

Initialize: \( S^0 = \{ \text{Indices of the } k \text{ largest coefficients of } \frac{Y^T \psi}{||\psi||_2} \} \),
\[
\beta = \text{argmin}_\beta ||\psi - Y\beta||_2^2 \text{ s.t. supp}\{\beta\} \subseteq S^0; \forall z, \beta_z \geq 0,
\]
\[
r^0 = \psi - Y\beta.
\]

Iterations:
1: for \( i = 1 \) to \( k \) do
2: \( \hat{S}^i = S^{i-1} \cup \{ \text{Indices of the } k \text{ largest coefficients of } \frac{Y^T r^{i-1}}{||r^{i-1}||_2} \} \).
3: \( \beta = \text{argmin}_\beta ||\psi - Y\beta||_2^2 \text{ s.t. supp}\{\beta\} \subseteq \hat{S}^i; \forall z, \beta_z \geq 0 \).
4: \( S^i = \{ \text{Indices of the largest } k \text{ coefficients of } \beta \} \).
5: \( \xi^* = \text{argmin}_{\xi^*} ||\psi - Y\xi^*||_2^2 \text{ s.t. supp}\{\xi^*\} \subseteq S^i; \forall z, \xi^*_z \geq 0 \).
6: \( r^i = \psi - Y\xi^* \).
7: if \( ||r^i||_2 = 0 \) or \( ||r^i||_2 \geq ||r^{i-1}||_2 \) then
8: break.
9: end if
10: end for
11: if \( ||r^i||_2 > ||r^{i-1}||_2 \) then
12: \( \xi^i = \text{argmin}_{\xi^i} ||\psi - Y\xi^i||_2^2 \text{ s.t. supp}\{\xi^i\} \subseteq S^{i-1}; \forall z, \xi^i_z \geq 0 \).
13: end if

Output:
14: Sparse codes \( \xi^* \).

In NSP, we iteratively identify a subspace of at-most \( k \) columns of \( Y \), such that the input signal \( \psi \) (or its best approximation) lies in the positive orthant of this subspace. Following the conventions of the sparse representation literature, below, we refer to \( Y \) as the dictionary and to its columns as the atoms. To initialize the dictionary, we first identify the indices of \( k \) atoms, having the smallest angles with \( \psi \). These indices are recorded in a set \( S^0 \). Then, we compute the orthogonal projection of \( \psi \) onto the positive orthant of the atoms indexed in \( S^0 \). In Algorithm 6, the
operator ‘\(\text{supp}\{\cdot\}\)’ used for this purpose, indicates the support, i.e. the indices of the non-zero coefficients, of a vector. We then compute the residue vector \(r^0\) by subtracting from \(\psi\) its computed projection (i.e. \(Y\beta\)).

The algorithm performs at most \(k\) iterations. In the \(i\)th iteration, it augments the set \(S^{i-1}\) with \(k\) more indices to compute a set \(\hat{S}^i\). The newly added indices correspond to the atoms subtending the smallest angles with the residue vector \(r^{i-1}\). Using \(\hat{S}^i\), the \(k\)-best atoms are identified in \(S^i\). These atoms maximally contribute to the non-negative least squares approximation of \(\psi\), when only the atoms in \(\hat{S}^i\) are used as the basis. This procedure is stated in lines ‘3-4’ of the algorithm. Next, \(\psi\) is approximated in the positive orthant of the column space of the atoms indexed in \(S^i\). This results in a potential solution \(\xi^*\), that is used to compute the new residue vector \(r^i\) (lines ‘5-6’). If the new residue is zero or it does not improve in the current iteration, the iterations stop and \(\xi^*\) becomes the solution. However, if the residue increases as a result of the \(i\)th iteration, \(S^{i-1}\) is used to re-compute the sparse codes \(\xi^*\), as mentioned in line ‘11-13’ of the algorithm.

We emphasize on one important property of NSP. That is, it reconstructs \(\psi\) using a non-negative linear combination of ‘at-most’ \(k\) pixels indexed in \(S\). Bruckstein et al. [31] showed that the non-negativty in such reconstructions naturally leads to the sparsest combinations of the signals. For NSP, this means that the sparsity of \(\xi^*\) is upper bounded by \(k\). If \(\psi\) can be constructed with fewer pixels, the algorithm automatically uses that sparsity level, to the end that it uses only one pixel if it belongs to a pure endmember. Thus, the pixels considered as pure endmembers are explicitly recorded by \(\Xi\) in our approach. It is easy to see that NSP can potentially compute the exact same solutions under different values of \(k\), if these values are above a certain threshold. This is because the non-negativity imposes sparsity by forcing the unrequired coefficients of \(\xi^*\) to zero [31]. However, for a larger \(k\), a larger subspace would have to be parsed for computing the solution, that can result in more computations. Therefore, smaller \(k\) values are preferred for our approach.

It is worth mentioning that Algorithm 6 differs from SP [60] in two core operations, namely; (a) identifying the sets \(S^0\) and \(\hat{S}^i\), and in (b) computing the vectors \(\beta\) and \(\xi^*\). In SP, \(S^0\) records the indices of the \(k\) coefficients of the vector \(s_0 = Y\psi\) that have the largest magnitudes. Similarly, \(\hat{S}^i\) is computed by augmenting \(S^{i-1}\) with the indices of the largest magnitude entries in \(s_i = Yr^{i-1}\). Such constructions of the sets are governed by the fact that SP also allows negative coefficients in \(\beta\) (and \(\xi^*\)). We place a non-negativity constraint over \(\beta\) (and \(\xi^*\)), allowing only those atoms in sets \(S^0\) and \(\hat{S}^i\) that have positive correlations with \(\psi\) and \(r^{i-1}\), respectively.
5.4 Experiments with synthetic data

To quantitatively analyze the performance of our approach we first experiment with synthetically mixed endmembers.

5.4.1 Data

To generate the synthetic data, we used the NASA Jet Propulsion Lab’s Advanced Space-borne Thermal Emission and Reflectance Radiometer (ASTER) library (http://speclib.jpl.nasa.gov). We selected a set of 25 spectra from the library, that were considered as pure endmembers in our experiments. Henceforth, we denote this set as $\mathcal{P}_{25}$. We randomly selected the endmembers in $\mathcal{P}_{25}$ such that their mutual coherence [196] was higher than 0.9995. Mutual coherence ($0 \leq \mu \leq 1$) is defined as the maximum absolute inner product between any two distinct endmembers in the set. It is well established that the accuracy of matrix factorization approaches can be adversely affected by the large mutual coherence of the sources generating the data [196], [6]. For $\mathcal{P}_{25}$, the value of $\mu = 0.9998$, which made our experiments challenging. In the selected set, 18 spectra belonged to minerals of different kinds and granularity levels, 5 spectra were of different rock types and one spectra belonged to a soil sample. To follow a general protocol [6], [102], the ASTER library spectra were used after resampling at the sampling wavelengths of the NASA’s AVIRIS sensor [80].

In a single experiment, we simulated a synthetic hyperspectral image with 10,000 mixed pixels, using a set $\mathcal{P}_{20} \subset \mathcal{P}_{25}$ that contained 20 randomly chosen endmembers. The final results were computed by averaging the performance on 10 images. Each pixel $y_j \in \mathbb{R}^m$ was constructed by mixing $2 \leq p \leq 5$ pure endmembers such that their fractional abundances followed a Dirichlet distribution. These settings are motivated by the common knowledge that the number of materials in a typical remote sensing scene is 20 or less [102], and that the number of mixed materials for a pixel is usually of the order of 4 to 5 [110]. Existence of a pure pixel gives our approach an additional advantage, as it then directly selects that pixel as an endmember. Nevertheless, pure pixels are not always present in real data. Therefore, we evaluate our approach for a more challenging scenario and keep $p > 1$ to ensure the absence of pure endmember pixels. To simulate the outliers, we corrupted 3% of the pixels of each image by randomly saturating 50% of their channels. This is done by replacing the reflectance values by 1 at those channels.

We have considered spectral mixing with both white and correlated additive noises. To simulate the white noise we used the Matlab’s inbuilt \texttt{awgn} function,
with measured signal power. We followed Bioucas-Dias and Nascimento [23] to add the correlated noise to the data. We defined the diagonal noise correlation matrix such that its entries formed a Gaussian shape. A diagonal entry \((\sigma^2_h)\) of the matrix was computed as:

\[
\sigma^2_h = \frac{\sum_{j=1}^{n} ||y_j||^2_2}{n \times 10^{(\frac{\text{SNR}}{10})} \eta \sqrt{2\pi}} \exp\left(\frac{(h-m)^2}{2\eta^2}\right) \forall h \in \{1, \ldots, m\},
\]

(5.14)

where \(\eta\) controls the variance of the bell curve. In our experiments we fixed \(\eta = 18\).

In Fig. 5.2, we illustrate the correlated noise added to a mixed pixel. For further theoretical details on generation of the correlated noise, we refer to [23].

### 5.4.2 Evaluation metrics

To evaluate the performance, we compute the average spectral angle \((\theta_{avg})\) in \(\mathbb{R}^m\) between our estimates of the endmembers and the actual endmembers used to generate the synthetic data. Suppose we used a set \(P = \{\varphi^*_q | \varphi^*_q \text{ is a pure endmember}\}\) to generate a synthetic image, then \(\theta_{avg}\) is defined as:

\[
\theta_{avg} = \frac{180}{\pi |P|} \sum_{q=1}^{\midP\mid} \arccos \frac{\varphi_q^T \varphi_q^*}{||\varphi_q||_2 ||\varphi_q^*||_2},
\]

(5.15)

where \(|.\)| denotes the cardinality of the set and \(\varphi_q\) is the estimated endmember, that best matches\(^3\) the \(q^{th}\) true endmember. Note that the computed angle is in degrees.

The metric \(\theta_{avg}\) only evaluates the soundness of the extracted endmembers. To evaluate the estimated fractional abundances, we used the Root Mean Squared Error

\(^3\)Matching is done by minimizing the angles between the estimated and the true endmembers, such that, for each true endmember, there is only one match and each estimated endmember is used only once in the process.
5.4. Experiments with synthetic data

(RMSE), defined as:

\[
RMSE = \sqrt{\frac{||A - A^*||_F^2}{|\mathcal{P}| \times n}},
\]

where \( A^* \in \mathbb{R}^{|\mathcal{P}| \times n} \) is the actual fractional abundance matrix, used to generate the image. We compute the RMSE after matching the computed endmembers with the actual endmembers and re-arranging the computed matrix \( A \) accordingly.

5.4.3 Benchmarking

To benchmark, we mainly compare our approach to the popular constrained matrix factorization methods. We also report the results of commonly used endmember extraction algorithms, combined with abundance estimation methods. We used the author-provided implementations for all the approaches. Among the sparse matrix factorization techniques, we compare our results with K-SVD [2] and the Online Dictionary Learning (ODL) [135]. Note that, the sparsity constrained matrix factorization has already established its effectiveness in hyperspectral unmixing [81], [103], [164], [44]; and K-SVD and ODL are the state-of-the-art approaches. We additionally compare our results with a variant of ODL that computes non-negative endmembers and fractional abundances [135]. We refer to this approach as ODL-NN. We also include the Archetypal Analysis based matrix factorization [59] in our comparisons. Similar to our approach, the Archetypal Analysis constructs the endmembers as non-negative linear combinations of the observed pixels. However, unlike our approach, it also constrains an endmember to lie in a simplex whose vertices are formed by the observed pixels. In our experiments, we used the Archetypal Analysis enhancement of Chen et al. [49] that also accounts for outliers. To explicate the effect of robustness against outliers in our approach, we also provide the results of the non-robust variant of our approach, denoted below as CMF. This variant is implemented by removing lines ‘3-6’ and the matrix \( \Delta \) from line ‘9’ in Algorithm 5.

Among the endmember extraction techniques, we compare the performance of our approach with the commonly used methods, known as Vector Component Analysis (VCA) [146], Successive Volume Maximization (SVMAX) [42] and its variant, Alternating Volume Maximization (AVMAX) [42]. The endmembers extracted by these approaches were used with abundance estimation methods to perform the unmixing. We used the SUUnSAL implementation [22] (as used by our approach) for computing the abundances. For VCA and AVMAX, we set the implementation to solve a fully constrained least squares problem. For SVMAX, we used the constrained version of SUUnSAL that additionally imposes sparsity on the abundance estimation.
vectors along ANC and ASC. In our experiments, these combinations resulted in the best performance of the endmember extraction based unmixing.

5.4.4 Results

We summarize the results of our experiments in Tables 5.1 and 5.2. In Table 5.1, we report the average spectral angles between the ground truth and the computed endmembers. These values were computed when the data contained white noise, correlated noise and no noise at all. For the noisy cases, we used SNR = 30dB. The corresponding RMSEs of the computed fractional abundance matrices are reported in Table 5.2. In our experiments, all the approaches required the value of the total number of endmembers ‘K’ as an input. Matching the cardinality of the set \( P_{25} \), we chose \( K = 25 \) for each technique. All the remaining parameters of the approaches were carefully optimized on separate cross validation data. For our approach, we used the sparsity level \( k = 5 \) and \( \varepsilon = 10^{-10} \). For K-SVD, we used 10 as the sparsity threshold. The regularization constant in ODL and ODL-NN was fixed to \( 10^{-6} \). We refer to the original works for details on the significance of these parameters.

From Table 5.1, we can see that the proposed RCMF is able to recover the endmembers very accurately, both in the presence and absence of noise. Compared to RCMF, its non-robust variant CMF generally underperforms in these settings. Nevertheless, its performance remains acceptable. When there is no noise AVMAX [42] and VCA [146] are unable to converge. This happens because of absence of pure pixels in the data. On the other hand, SVMAX [42] is able to recover the endmembers with an accuracy higher than RCMF. However, it happens only for the noise-free case. For practical settings, RCMF outperforms SVMAX with a significant margin. In Fig. 5.3, we show a few representative examples of the endmembers recovered by RCMF and CMF along the ground truth.

Due to the better recovery of the endmembers, RCMF is also able to compute their fractional abundances accurately. This is evident from Table 5.2. While computing the RMSEs, we disregarded the abundances related to the pixels representing the outliers. Therefore, the difference between the results of CMF and RCMF is not significant. Nevertheless, RCMF still outperforms CMF because of its better endmember recovery. One interesting observation in Table 5.2 is that the additional non-negativity constraint in ODL proved to be particularly effective in computing the abundances. Whereas the endmembers recovered by K-SVD and ODL were generally observed to be non-negative in our experiments, this was not the case for their computed abundances, unless the non-negativity was explicitly imposed. Due
5.4. Experiments with synthetic data

Table 5.1: Average spectral angle ($\theta_{avg}$): For noisy cases, the SNR is 30dB.

<table>
<thead>
<tr>
<th>Method</th>
<th>White noise</th>
<th>Correlated noise</th>
<th>Noise free</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCA [146]</td>
<td>10.13 ± 1.22</td>
<td>9.25 ± 0.58</td>
<td>–</td>
</tr>
<tr>
<td>AVMAX [42]</td>
<td>10.35 ± 1.50</td>
<td>9.34 ± 0.57</td>
<td>–</td>
</tr>
<tr>
<td>SVMAX [42]</td>
<td>10.55 ± 1.63</td>
<td>8.99 ± 0.79</td>
<td>3.54 ± 0.21</td>
</tr>
<tr>
<td>K-SVD[2]</td>
<td>12.85 ± 0.25</td>
<td>13.86 ± 0.70</td>
<td>7.40 ± 1.39</td>
</tr>
<tr>
<td>ODL[135]</td>
<td>10.65 ± 0.33</td>
<td>8.79 ± 0.58</td>
<td>7.70 ± 0.51</td>
</tr>
<tr>
<td>ODL-NN[135]</td>
<td>10.63 ± 0.29</td>
<td>8.79 ± 0.58</td>
<td>7.70 ± 0.51</td>
</tr>
<tr>
<td>Archetypal[49]</td>
<td>5.49 ± 0.36</td>
<td>4.95 ± 0.33</td>
<td>4.24 ± 0.48</td>
</tr>
<tr>
<td>CMF</td>
<td>5.35 ± 0.20</td>
<td>5.11 ± 0.25</td>
<td>4.11 ± 0.21</td>
</tr>
<tr>
<td>RCMF</td>
<td>5.19 ± 0.22</td>
<td>4.55 ± 0.59</td>
<td>3.97 ± 0.22</td>
</tr>
</tbody>
</table>

Table 5.2: RMSE of abundance matrices: For noisy cases, the SNR is 30dB.

<table>
<thead>
<tr>
<th>Method</th>
<th>White noise</th>
<th>Correlated noise</th>
<th>Noise free</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCA [146]</td>
<td>0.130 ± 8.8 × 10^{-3}</td>
<td>0.128 ± 4.08 × 10^{-3}</td>
<td>–</td>
</tr>
<tr>
<td>AVMAX [42]</td>
<td>0.129 ± 4.3 × 10^{-3}</td>
<td>0.127 ± 6.53 × 10^{-3}</td>
<td>–</td>
</tr>
<tr>
<td>SVMAX [42]</td>
<td>0.141 ± 1.1 × 10^{-2}</td>
<td>0.132 ± 7.9 × 10^{-3}</td>
<td>0.088 ± 8.08 × 10^{-3}</td>
</tr>
<tr>
<td>K-SVD[2]</td>
<td>2.075 ± 0.16</td>
<td>2.388 ± 1.419</td>
<td>6.06 ± 2.63</td>
</tr>
<tr>
<td>ODL[135]</td>
<td>1.506 ± 0.16</td>
<td>1.786 ± 0.200</td>
<td>88.98 ± 70.23</td>
</tr>
<tr>
<td>ODL-NN[135]</td>
<td>1.059 ± 0.07</td>
<td>1.261 ± 1.159</td>
<td>1.49 ± 0.05</td>
</tr>
<tr>
<td>Archetypal[49]</td>
<td>0.103 ± 2.2 × 10^{-3}</td>
<td>0.103 ± 6.43 × 10^{-3}</td>
<td>0.105 ± 4.00 × 10^{-3}</td>
</tr>
<tr>
<td>CMF</td>
<td>0.101 ± 0.8 × 10^{-3}</td>
<td>0.100 ± 0.010</td>
<td>0.106 ± 2.34 × 10^{-3}</td>
</tr>
<tr>
<td>RCMF</td>
<td>0.097 ± 3.5 × 10^{-5}</td>
<td>0.095 ± 6.6 × 10^{-3}</td>
<td>0.104 ± 1.10 × 10^{-2}</td>
</tr>
</tbody>
</table>
Figure 5.3: Examples of the endmembers recovered by the proposed approach at 35dB SNR.

Figure 5.4: Average spectral angle $\theta_{\text{avg}}$ as a function of the number of extracted endmembers $K$. The images are created using 20 pure spectra.

to better recovery of the endmembers by SVMAX [42] in the noise-free settings, constrained version of SUnSAL [22] resulted in the best abundance estimation when used with SVMAX in the absence of noise.

Generally, the total number of endmembers present in a scene is not known a priori. Therefore, we also evaluated the performance of approaches by varying the number endmembers allowed to be extracted by them. That is, by changing the number of the basis vectors allowed to be learned in the matrix factorization procedure. In Fig. 5.4, we plot $\theta_{\text{avg}}$ against this number. It can be seen that the approaches are generally able to extract the endmembers better by allowing more basis vectors. The value of $\theta_{\text{avg}}$ remains very small for our approach throughout the plots. Note that, we used 20 endmembers to generate a synthetic image. The combination of endmember extraction algorithm and abundance estimation algorithms performed somewhat similar to ODL [135] and ODL-NN, that use $\ell_1$-sparsity con-
5.4. Experiments with synthetic data

Figure 5.5: RMSE as a function of the number of extracted endmembers \( K \). The images are created with 20 pure spectra.

The legend only shows the names of the endmember extraction algorithms. Since the endmembers computed by ODL were generally non-negative, ODL-NN did not result in improved \( \theta_{\text{avg}} \). The performance of Archetypal Analysis is similar to our approach due to the similarities in the underlying assumptions of the algorithms.

The RMSE values of the fractional abundances for the above experiments are plotted in Fig. 5.5. For clarity, we do not plot the results of ODL, ODL-NN and KSVD in the figure because they were not comparable to the other results, as can be verified from Table 5.2. From Fig. 5.5, a consistent performance of the proposed approach is evident.

In Fig. 5.6 and 5.7 we plot the \( \theta_{\text{avg}} \) and RMSE values for the approaches against different levels of noise in the data. We extracted 25 basis vectors with each approach in these experiments. The plots clearly show that the proposed approach performs reasonably well even for very low SNR, and the performance generally improves with higher SNR. In Fig. 5.7, the RMSE values for KSVD, ODL and ODL-NN are not included because they were not comparable to the other results. Despite good results of SMAX in Tables 5.1 and 5.2, its performance did not improve too much at high SNR values. This happened due to the presence of outliers, which were not considered in the noise-free settings of Tables 5.1 and 5.2.

In all experiments, we used 300 iterations of K-SVD, ODL and ODL-NN. For Archetypal Analysis, we used 200 iterations. Our algorithms generally required less than 100 iterations for convergence. Nevertheless, we used \( Q = 100 \) iterations in our experiments. The total number of iterations for each algorithm was decided with the help of cross validation data. The mean computation time for all the experiments in Tables 5.1 and 5.2 is reported in Table 5.3. The time is computed on a desktop computer with Intel Core i7-2600 CPU @ 3.4GHz and 8 GB RAM. In the table,
the mean computation time of CMF exceeds RCMF because the later often reaches the breaking condition of NSP in line ‘7’ of Algorithm 6 in fewer iterations. In our opinion, this is a result of considering robustness against the outliers in RCMF. For the endmember extraction algorithms, the time is provided for the complete process of extracting the endmembers and computing the abundances. Compared to the proposed approach, these timings are better. Nevertheless, our approach does not assume presence of pure endmembers in the image and performs significantly better than these algorithms in practical conditions.

Table 5.3: Mean computation time in seconds, for unmixing one thousand pixels.

<table>
<thead>
<tr>
<th>Method</th>
<th>K-SVD</th>
<th>ODL</th>
<th>ODL-NN</th>
<th>Archetypal</th>
<th>VCA</th>
<th>AVMAX</th>
<th>SVMAX</th>
<th>CMF</th>
<th>RCMF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>18.09</td>
<td>18.76</td>
<td>18.81</td>
<td>17.82</td>
<td>3.27</td>
<td>3.18</td>
<td>3.20</td>
<td>18.13</td>
<td>17.67</td>
</tr>
</tbody>
</table>

Figure 5.6: Average spectral angle ‘θavg’ as a function of Signal to Noise Ratio (SNR). In the experiments, $K = 25$.

Figure 5.7: RMSE as a function of SNR. In the experiments, $K = 25$. 
5.5 Experiments with real data

We analyze the NASA’s AVIRIS Cuprite data (http://aviris.jpl.nasa.gov/data/free_data.html) with our approach to establish its effectiveness on a real-world unmixing problem. Since the quantitative evaluation of the results is not possible because of the unavailability of the ground truth, we present our results for qualitative analysis, following [102], [6]. The said data is collected by the AVIRIS sensor [80] over a region of Cuprite mines in Nevada. The region is well-studied for its geological properties that makes its hyperspectral image a suitable benchmark. The analyzed image is a 512×512×224 cube, acquired in the wavelength range 370-2500nm. To process the cube, we first removed its 36 channels corresponding to the wavelengths 370, 380, 1330 to 1430, 1780 to 1970, 2490 and 2500 nm. This is a common protocol [102], [6] to avoid the low SNR and water absorption bands in the analysis. Then, we selected a 350×350×188 sub-cube, where discernible spatial patterns of multiple minerals were present.

In Fig. 5.8, we display the fractional abundance maps computed by the proposed RCMF (second row) for the selected sub-cube. The maps are of five different minerals that were commonly analyzed by recent unmixing approaches [102], [6]. These minerals show clear spatial patterns in the analyzed region. For reference, we also provide the mineral classification maps in the first row of the figure. These maps were computed by the USGS Tricorder algorithm\(^4\) in year 1995. Although, they were computed two years prior to the acquisition of the analyzed hyperspectral image, these maps provide a good reference for the abundance maps because we can expect high abundance values at the pixels classified by these maps as the pure minerals [6]. We can also expect high proportions of the same minerals in the nearby regions of the pure pixels. It is clear from the figure that RCMF generally assigns large fractional abundances to the correct regions. The spatial patterns of the computed abundances clearly match the classification maps. For comparison, we also provide the abundance maps computed by solving a fully constrained least squares problem using the endmembers extracted by VCA [146]. These maps are shown in the third row of the figure. Very similar maps resulted when we used the CSUnSAL algorithm [22] for computing the abundances with VCA endmembers. Those maps are not included to avoid redundancy. In the fourth row, we provide the sparse unmixing results of CSUnSAL [22], when ASTER library spectra were used as endmembers. Note that, sparse unmixing is the state-of-the-art supervised unmixing framework and the accuracy of CSUnSAL is well-established for this framework.

Figure 5.8: Fractional abundance analysis of the AVIRIS data: The first row shows the classification maps by the Tricorder software. Results of RCMF are displayed in the second row. The third row shows the result of VCA endmembers used for solving a fully constrained least squares problem and the fourth row shows CSUnSAL [22] results with ASTER library spectra.

[102], [6], [100]. Nevertheless, the spatial patterns of the abundance maps computed by RCMF visually appear better than those computed by CSUnSAL. Interestingly, the maps resulting from VCA endmembers are very close to those computed by our approach. We observed similar resemblance in the maps of other minerals as well for the two approaches. This happens because our approach also solves a fully constraint least squares problem in line 2 of Algorithm 5 to compute the abundances. When RCMF learns the endmembers that are similar to those extracted by VCA, their abundances computed (simultaneously) by RCMF naturally resemble to those computed by the abundance estimation methods used in conjunction with VCA (or other endmember extraction algorithms).

In Fig. 5.9, we show the spectra of the minerals recovered by RCMF and VCA. For reference, we also plot the spectra from the ASTER library. The spectra are shown as normalized reflectance values plotted against the wavelengths. Note that, the ASTER Library spectra are only provided as a reference and should not be considered as the ground truth because they are measured in laboratory conditions,
5.6 Conclusion

We proposed a novel matrix factorization approach for linear hyperspectral unmixing. In addition to accounting for the non-negativity of the endmembers and the physical constraints over their abundances, the proposed approach forces the extracted endmembers to be sparse non-negative combinations of the observed pixels. The association between the pixels and the endmembers is explicitly noted by our approach. Moreover, our approach also incorporates robustness against any possible outliers among the pixels. That makes our results more reliable. We systematically designed an efficient matrix factorization algorithm for our approach. Our experiments with synthetic hyperspectral images corrupted by white and correlated noise quantitatively establish the effectiveness of our approach. The proposed approach also shows promising qualitative results on the real hyperspectral data.
PART II

Hyperspectral
Super-Resolution
CHAPTER 6

Sparse Spatio-spectral Representation for Hyperspectral Image Super-Resolution

Abstract

Existing hyperspectral imaging systems produce low spatial resolution images due to hardware constraints. We propose a sparse representation based approach for hyperspectral image super-resolution. The proposed approach first extracts distinct reflectance spectra of the scene from the available hyperspectral image. Then, the signal sparsity, non-negativity and the spatial structure in the scene are exploited to explain a high-spatial but low-spectral resolution image of the same scene in terms of the extracted spectra. This is done by learning a sparse code with an algorithm G-SOMP+. Finally, the learned sparse code is used with the extracted scene spectra to estimate the super-resolution image. Comparison of the proposed approach with the state-of-the-art methods on both ground-based and remotely-sensed public hyperspectral image databases shows that the presented method generally achieves the lowest error rate on the test images of the three datasets.

Keywords: Hyperspectral, super-resolution, spatio-spectral, sparse representation.

6.1 Introduction

Hyperspectral imaging acquires a faithful representation of the scene radiance by integrating it against several basis functions that are well localized in the spectral domain. The spectral characteristics of the resulting representation have proven critical in numerous applications, ranging from remote sensing [24], [4] to medical imaging [109]. They have also been reported to improve the performance in computer vision tasks, such as, tracking [152], segmentation [192], recognition [201] and document analysis [111]. However, contemporary hyperspectral imaging lacks severely in terms of spatial resolution [109], [91]. The problem stems from the fact that each spectral image acquired by a hyperspectral system corresponds to a very narrow spectral window. Thus, the system must use long exposures to collect enough photons to maintain a good signal-to-noise ratio of the spectral images. This results in low spatial resolution of the hyperspectral images.

Published in Proc. of European Conference on Computer Vision (ECCV), 2014.
Normally, spatial resolution can be improved with high resolution sensors. However, this solution is not too effective for hyperspectral imaging, as it further reduces the density of the photons reaching the sensor. Keeping in view the hardware limitations, it is highly desirable to develop software based techniques to enhance the spatial resolution of hyperspectral images. In comparison to the hyperspectral systems, the low spectral resolution imaging systems (e.g. RGB cameras) perform a gross quantization of the scene radiance - losing most of the spectral information. However, these systems are able to preserve much finer spatial information of the scenes. Intuitively, images acquired by these systems can help in improving the spatial resolution of the hyperspectral images.

This work develops a sparse representation \[156\] based approach for hyperspectral image super-resolution, using a high-spatial but low-spectral resolution image (henceforth, only called the *high spatial resolution image*) of the same scene. The proposed approach uses the hyperspectral image to extract the reflectance spectra related to the scene. This is done by solving a constrained sparse representation problem using the hyperspectral image as the input. The basis formed by these spectra is transformed according to the spectral quantization of the high spatial resolution image. Then, the said image and the transformed basis are fed to a simultaneous sparse approximation algorithm G-SOMP+. Our algorithm is a generalization of Simultaneous Orthogonal Matching Pursuit (SOMP) \[199\] that additionally imposes a non-negativity constraint over its solution space. Taking advantage of the spatial structure in the scene, G-SOMP+ efficiently learns a sparse code. This sparse code is used with the reflectance spectra of the scene to estimate the super-resolution hyperspectral image. We test our approach using the hyperspectral images of objects, real-world indoor and outdoor scenes and remotely sensed hyperspectral image. Results of the experiments show that the proposed approach consistently performs better than the existing methods on all the data sets.

This paper is organized as follows. Section 6.2 reviews the previous literature related to the proposed approach. We formalize our problem in Section 6.3. The proposed solution is described in Section 6.4 of the paper. In Section 6.5, we give the results of the experiments that have been performed to evaluate the approach. We dedicate Section 6.6 for the discussion on the results and the parameter settings. The paper concludes with a brief summary in Section 6.7.

### 6.2 Related Work

Hardware limitations have lead to a notable amount of research in software based techniques for high spatial resolution hyperspectral imaging. The software based ap-
proaches that use image fusion [209] as a tool, are particularly relevant to our work. Most of these approaches have originated in the remote sensing literature because of the early introduction of hyperspectral imaging in the airborne/spaceborne observatory systems. In order to enhance the spatial resolution of the hyperspectral images these approaches usually fuse a hyperspectral image with a high spatial resolution pan-chromatic image. This process is known as pan-sharpening [10]. A popular technique ([37], [3], [96], [116]) uses a linear transformation of the color coordinates to improve the spatial resolution of hyperspectral images. Exploiting the fact that human vision is more sensitive to luminance, this technique fuses the luminance component of a high resolution image with the hyperspectral image. Generally, this improves the spatial resolution of the hyperspectral image, however the resulting image is sometimes spectrally distorted [39].

In spatio-spectral image fusion, one class of methods exploits unmixing ([144], [243]) for improving the spatial resolution of the hyperspectral images. These methods only perform well for the cases when the spectral resolutions of the two images are not too different. Furthermore, their performance is compromised in highly mixed scenarios [91]. Zurita-Milla et al. [244] employed a sliding window strategy to mitigate this issue. Image filtering is also used for interpolating the spectral images to improve the spatial resolution [115]. In this case, the implicit assumption of smooth spatial patterns in the scenes often produces overly smooth images.

More recently, matrix factorization has played an important role in enhancing the spatial resolution of the ground based and the remote sensing hyperspectral imaging systems ([109], [91], [215], [227]). Kawakami et al. [109] have proposed to fuse a high spatial resolution RGB image with a hyperspectral image by decomposing each of the two images into two factors and constructing the desired image from the complementary factors of the two decompositions. A very similar technique has been used by Huang et al. [91] for remote sensing data. The main difference between [109] and [91] is that the latter uses a spatially down-sampled version of the high spatial resolution image in the matrix factorization process. Wycoff et al. [215] have proposed an algorithm based on Alternating Direction Method of Multipliers (ADMM) [30] for the factorization of the matrices and later using it to fuse the hyperspectral image with an RGB image. Yokoya et al. [227] have proposed a coupled matrix factorization approach to fuse multi-spectral and hyperspectral remote sensing images to improve the spatial resolution of the hyperspectral images.

The matrix factorization based methods are closely related to our approach. However, our approach has major differences with each one of them. Contrary to
these methods, we exploit the spatial structure in the high spatial resolution image for the improved performance. The proposed approach also takes special care of the physical significance of the signals and the processes related to the problem. This makes our formalization of the problem and its solution unique. We make use of the non-negativity of the signals, whereas [109] and [91] do not consider this notion at all. In [215] and [227] the authors do consider the non-negativity of the signals, however their approaches require the down sampling matrix that converts the high resolution RGB image to the corresponding bands of the low resolution hyperspectral image. Our approach does not impose any such requirement.

6.3 Problem Formulation

We seek estimation of a super-resolution hyperspectral image \( S \in \mathbb{R}^{M \times N \times L} \), where \( M \) and \( N \) denote the spatial dimensions and \( L \) represents the spectral dimension, from an acquired hyperspectral image \( Y_h \in \mathbb{R}^{m \times n \times L} \) and a corresponding high spatial (but low spectral) resolution image of the same scene \( Y \in \mathbb{R}^{M \times N \times l} \). For our problem, \( m \ll M, n \ll N \) and \( l \ll L \), which makes the problem severely ill-posed. We consider both of the available images to be linear mappings of the target image:

\[
Y = \Psi(S), \quad Y_h = \Psi_h(S),
\]

(6.1)

where \( \Psi : \mathbb{R}^{M \times N \times L} \rightarrow \mathbb{R}^{M \times N \times l} \) and \( \Psi_h : \mathbb{R}^{M \times N \times L} \rightarrow \mathbb{R}^{m \times n \times L} \).

A typical scene of the ground based imagery as well as the space-borne/airborne imagery contains only a small number of distinct materials [9], [110]. If the scene contains \( q \) materials, the linear mixing model (LMM) [102] can be used to approximate a pixel \( y_h \in \mathbb{R}^L \) of \( Y_h \) as

\[
y_h \approx \sum_{\omega=1}^{c} \varphi_\omega \alpha_\omega, \quad c \leq q,
\]

(6.2)

where \( \varphi_\omega \in \mathbb{R}^L \) denotes the reflectance of the \( \omega \)-th distinct material in the scene and \( \alpha_\omega \) is the *fractional abundance* (i.e. proportion) of that material in the area corresponding to the pixel. We rewrite (6.2) in the following matrix form:

\[
y_h \approx \Phi \alpha.
\]

(6.3)

In (6.3), the columns of \( \Phi \in \mathbb{R}^{L \times c} \) represent the reflectance vectors of the underlying materials and \( \alpha \in \mathbb{R}^c \) is the coefficient vector. Notice that, when the scene represented by a pixel \( y_h \) also includes the area corresponding to a pixel \( y \in \mathbb{R}^l \) of
6.4 Proposed Solution

Let $\mathcal{D}$ be a finite collection of unit-norm vectors in $\mathbb{R}^L$. In our settings, $\mathcal{D}$ is the dictionary whose elements (i.e. the atoms) are denoted by $\varphi_\omega$, where $\omega$ ranges over an index set $\Omega$. More precisely, $\mathcal{D} = \{ \varphi_\omega : \omega \in \Omega \} \subset \mathbb{R}^L$. Considering (6.3)-(6.5), we are interested in forming the matrix $\Phi$ from $\mathcal{D}$, such that

$$\bar{Y}_h \approx \Phi A,$$

(6.6)

where $\bar{Y}_h \in \mathbb{R}^{L \times mn}$ is the matrix formed by concatenating the pixels of the hyperspectral image $Y_h$ and $A$ is the coefficient matrix with $\alpha_i$ as its $i^{th}$ column. We propose to draw $\Phi$ from $\mathbb{R}^{L \times k}$, such that $k > q$; see (6.2). This is because, the LMM in (6.2) approximates a pixel assuming linear mixing of the material reflectances. In the real world, phenomena like multiple light scattering and existence of intimate material mixtures also cause non-linear mixing of the reflectance [102]. This usually alters the reflectance spectrum of a material or results in multiple distinct reflectance spectra of the same material in the scene. The matrix $\Phi$ must also account for these spectra. Henceforth, we use the term dictionary for the matrix $\Phi$.

According to the model in (6.6), each column of $\bar{Y}_h$ is constructed using a very small number of dictionary atoms. Furthermore, the atoms of the dictionary are non-negative vectors as they correspond to reflectance spectra. Therefore, we propose to

---

\(^1\)Formally, $\Phi$ is the dictionary synthesis matrix [199]. However, we follow the convention of the previous literature in dictionary learning (e.g. [2], [198]), which rarely distinguishes the synthesis matrix from the dictionary.
solve the following constrained sparse representation problem to learn the proposed
dictionary $\Phi$:
\[
\min_{\Phi,A} ||A||_1 \text{ s.t. } ||\bar{Y}_h - \Phi A||_2 \leq \eta, \quad \varphi_\omega \geq 0, \quad \forall \omega \in \{1, ..., k\}, \quad (6.7)
\]
where $||.||_1$ and $||.||_2$ denote the element-wise $l_1$ norm and the Euclidean norm of
the matrices respectively, and $\eta$ represents the modeling error. To solve (6.7) we
use the online dictionary learning approach proposed by Mairal et al. [135] with an
additional non-negativiity constraint on the dictionary atoms - we refer the reader
to the original work for details.

Once $\Phi$ is known, we must compute an appropriate coefficient matrix $B \in \mathbb{R}^{k \times MN}$; as suggested by (6.5), to estimate the target image $S$. This matrix is
computed using the learned dictionary and the image $Y$ along with two important
pieces of prior information. a) In the high spatial resolution image, nearby pixels
are likely to represent the same materials in the scene. Hence, they should be well
approximated by a small group of the same dictionary atoms. b) The elements of $B$
must be non-negative quantities because they represent the fractional abundances
of the spectral signal sources in the scene. It is worth mentioning that we could
also use (b) for $A$ in (6.7), however, there we were interested only in $\Phi$. Therefore,
a non-negativity constraint over $A$ was unnecessary. Neglecting this constraint in
(6.7) additionally provides computational advantages.

Considering (a), we process the image $Y$ in terms of small disjoint spatial patches
for computing the coefficient matrix. We denote each of the image patch by $P \in \mathbb{R}^{M_P \times N_P \times l}$ and estimate its corresponding coefficient matrix $B_P \in \mathbb{R}^{k \times M_P N_P}$ by
solving the following constrained simultaneous sparse approximation problem:
\[
\min_{B_P} ||B_P||_{\text{row},0} \text{ s.t. } ||\bar{P} - \tilde{\Phi} B_P||_2 \leq \varepsilon, \quad \beta_{p_i} \geq 0 \quad \forall i \in \{1, ..., M_P N_P\}, \quad (6.8)
\]
where $\bar{P} \in \mathbb{R}^{l \times M_P N_P}$ is formed by concatenating the pixels in $P$, $\tilde{\Phi} \in \mathbb{R}^{l \times k}$ is the
transformed dictionary $\tilde{\Phi} = T \Phi$; see (6.4), and $\beta_{p_i}$ denotes the $i^{th}$ column of the ma-
trix $B_p$. In the above objective function, $||.||_{\text{row},0}$ denotes the row-$l_0$ quasi-norm[199]
of the matrix, which represents the cardinality of its row-support\footnote{Set of indices for the non-zero rows of the matrix.}. Formally,
\[
||B_P||_{\text{row},0} = \left| \bigcup_{i=1}^{M_P N_P} \text{supp}(\beta_{p_i}) \right|,
\]
where supp(.) indicates the support of a vector and $|.|$ denotes the cardinality of
a set. Tropp [198] has argued that (6.8) is an NP-hard problem without the non-
negativity constraint. The combinatorial complexity of the problem does not change
with the non-negativity constraint over the coefficient matrix. Therefore, the problem must either be relaxed [198] or solved by the greedy pursuit strategy [199]. We prefer the latter because of its computational advantages [31] and propose a simultaneous greedy pursuit algorithm, called G-SOMP+, for solving (6.8). The proposed algorithm is a generalization of a popular greedy pursuit algorithm Simultaneous Orthogonal Matching Pursuit (SOMP) [199], which additionally constrains the solution space to non-negative matrices. Hence, we denote it as G-SOMP+. Here, the notion of ‘generalization’ is similar to the one used in [205] that allows selection of multiple dictionary atoms in each iteration of Orthogonal Matching Pursuit (OMP) [197] to generalize OMP.

G-SOMP+ is given below as Algorithm 7. The algorithm seeks an approximation of the input matrix \( \bar{P} \) - henceforth, called the patch - by selecting the dictionary atoms \( \tilde{\phi}_\xi \) indexed in a set \( \Xi \subset \Omega \), such that, \( |\Xi| \ll |\Omega| \) and every \( \tilde{\phi}_\xi \) contributes to the approximation of the whole patch. In its \( i^{th} \) iteration, the algorithm first computes the cumulative correlation of each dictionary atom with the residue of its current approximation of the patch (line 5 in Algorithm 7) - the patch itself is considered as the residue for initialization. Then, it identifies \( L \) (an algorithm parameter) dictionary atoms with the highest cumulative correlations. These atoms are added to a subspace indexed in a set \( \Xi_i \), which is empty at initialization. The aforementioned subspace is then used for a non-negative least squares approximation of the patch (line 8 in Algorithm 7) and the residue is updated. The algorithm stops if the updated residue is more than a fraction \( \gamma \) of the residue in the previous iteration. Note that, the elements of the set \( \Xi \) in G-SOMP+ also denote the row-support of the coefficient matrix. This is because, a dictionary atom can only participate in the patch approximation if the corresponding row of the coefficient matrix has some non-zero element in it. G-SOMP+ has three major differences from SOMP. 1) Instead of integrating the absolute correlations, it sums the correlations between a dictionary atom and the residue vectors (line 5 of Algorithm 7). 2) It approximates the patch in each iteration with the non-negative least squares method, instead of using the standard least squares approximation. 3) It selects \( L \) dictionary atoms in each iteration instead of a single dictionary atom. In the above mentioned difference, (1) and (2) impose the non-negativity constraint over the desired coefficient matrix. On the other hand, (3) primarily aims at improving the computation time of the algorithm. G-SOMP+ also uses a different stopping criterion than SOMP, that is controlled by \( \gamma \) - the residual decay parameter. We defer further discussion on (3) and the stopping criterion to Section 6.6. G-SOMP+...
Algorithm 7 G-SOMP+

Initialization:
1: Iteration: $i = 0$
2: Initial solution: $B^0 = 0$
3: Initial residue: $R^0 = \tilde{P} - \tilde{\Phi}B^0 = \tilde{P}$
4: Initial index set: $\Xi^0 = \emptyset = \text{row-supp}\{B^0\}$, row-supp$\{B\} = \{1 \leq t \leq k : \beta^t \neq 0\}$, where $\beta^t$ is the $t^{th}$ row of $B$.

Main Iteration: Update iteration: $i = i + 1$
5: Compute $b_j = \sum_{\tau = 1}^{M_P N_P} \tilde{\Phi}_\tau b_j T \tilde{R}^{i-1}, \forall j \in \{1, ..., k\}$, where, $X_z$ denotes the $z^{th}$ column of the matrix $X$.
6: $\Xi^i = \Xi^{i-1} \cup \mathcal{N}$
7: $B^i = \min ||\tilde{\Phi}B^i - \tilde{P}||_2$ s.t. row-supp$\{B\} = \Xi^i, \beta^t \geq 0, \forall t$
8: $R^i = \tilde{P} - \tilde{\Phi}B^i$
9: If $||R^i||_2 > \gamma||R^{i-1}||_2$ stop, otherwise iterate again.

has been proposed specifically to solve the constrained simultaneous sparse approximation problem in (6.8). Therefore, it is able to approximate a patch better than a generic greedy pursuit algorithm (e.g. SOMP).

Solving (6.8) for each image patch results in the desired coefficient matrix $B$ that is used with $\Phi$ to compute $\hat{\bar{S}} \in \mathbb{R}^{L \times MN}$, which is the estimate of the super-resolution hyperspectral image $\bar{S} \in \mathbb{R}^{L \times MN}$ (in matrix form).

$$\hat{\bar{S}} = \Phi B$$ (6.9)

Fig. 6.1 pictorially summarizes the proposed approach.

6.5 Experimental Results

We have evaluated our approach\textsuperscript{3} using ground based hyperspectral images as well as remotely sensed data. For the ground based images, we have conducted experiments with two different public databases. The first database \textsuperscript{[226]}, called the CAVE database, consists of 32 hyperspectral images of everyday objects. The 512 × 512 spectral images of the scenes are acquired at a wavelength interval of 10 nm in the range 400 – 700 nm. The second is the Harvard database \textsuperscript{[40]}, which consists of hyperspectral images of 50 real-world indoor and outdoor scenes. The

\textsuperscript{3}Source code available at http://www.csse.uwa.edu.au/~ajmal/code/HSISuperRes.zip
6.5. Experimental Results

Figure 6.1: Schematic of the proposed approach: The low spatial resolution hyperspectral (HS) image is used for learning a dictionary whose atoms represent reflectance spectra. This dictionary is transformed and used with the high-spatial but low-spectral resolution image to learn a sparse code by solving a constrained simultaneous sparse approximation problem. The sparse code is used with the original dictionary to estimate the super-resolution HS image.

1392 × 1040 spectral images are sampled at every 10 nm from 420 to 720 nm. Hyperspectral images of the databases are considered as the ground truth for the super-resolution hyperspectral images. We down-sample a ground truth image by averaging over 32 × 32 disjoint spatial blocks to simulate the low spatial resolution hyperspectral image $Y_h$. From the Harvard database, we have only used 1024 × 1024 image patches to match the down-sampling strategy. Following [215], a high spatial (but low spectral) resolution image $Y$ is created by integrating a ground truth image over the spectral dimension, using the Nikon D700 spectral response\(^4\) - which makes $Y$ a simulated RGB image of the same scene. Here, we present the results on eight representative images from each database, shown in Fig. 6.2. We have selected

\(^4\)https://www.maxmax.com/spectral_response.htm
Chapter 6. Sparse Representation for HSI Super-Resolution

Figure 6.2: RGB images from the databases. First row: Images from the CAVE database [226]. Second row: Images from the Harvard database [40].

these images based on the variety of the scenes. For our experiments, we initialize the dictionary $\Phi$ with random pixels from $Y_h$. Thus, the inherent smoothness of the pixels serves as an implicit loose prior on the dictionary.

Fig. 6.3 shows the results of using our approach for estimating the super-resolution hyperspectral images of ‘Painting’ and ‘Peppers’ (see Fig. 6.2). The top row shows the input $16 \times 16$ hyperspectral images at 460, 540 and 620 nm. The ground truth images at these wavelengths are shown in the second row, which are clearly well approximated in the estimated images shown in the third row. The fourth row of the figure shows the difference between the ground truth images and the estimated images. The results demonstrate a successful estimation of the super-resolution spectral images. Following the protocol of [109] and [215], we have used Root Mean Square Error (RMSE) as the metric for further quantitative evaluation of the proposed approach and its comparison with the existing methods. Table 6.1\(^5\) shows the RMSE values of the proposed approach and the existing methods for the images of the CAVE database [226]. Among the existing approaches we have chosen the Matrix Factorization method (MF) in [109], the Spatial and Spectral Fusion Model (SASFM) [91], the ADMM based method [215] and the Coupled Matrix Factorization method (CMF) [227] for the comparison. Most of these matrix factorization based approaches have been shown to outperform the other techniques discussed in Section 6.2. To show the difference in the performance, Table 6.1 also includes some results from the Component Substitution Method (CSM) [3] - taken directly

\(^5\)Please note that the results reported in this Table and Table 6.2 are slightly different from the original publication [8] due to correction of a minor error in the evaluation metric used in the original publication. We also note that Lanaras et al. [120] also claimed different performance of CMF [227] on the same images after publication of this work. We refer to [227] for the details on the use of CMF to achieve these results. The results reported in this work are directly taken from literature [215].
Figure 6.3: Spectral images for Painting (Left) and Peppers (Right) at 460, 540 and 620 nm. Top row: 16 × 16 low resolution hyperspectral (HS) images. Second row: 512 × 512 ground truth images. Third row: Estimated 512 × 512 HS images. Fourth row: Corresponding error images, where the scale is in the range of 8 bit images.

from [109]. We have used our own implementations of MF and SASFM because of unavailability of the public codes from the authors. To ensure an un-biased comparison, we take special care that the results achieved by our implementations are at least as good as the results reported originally by the authors on the same images. The results of CSM and ADMM are taken directly from [215]. Note that, these algorithms also require a priori knowledge of the spatial transform between the hyperspectral image and the high resolution image, because of which they are highlighted in red in the table. We have also experimented by replacing G-SOMP+ in the our approach with SOMP; its non-negative variant SOMP+ and its generalization G-SOMP. The means of the RMSEs computed over the complete CAVE database are 6.37, 6.21 and 5.32 when we replace G-SOMP+ with SOMP, SOMP+ and G-SOMP respectively, in our approach. This value is 3.64 for G-SOMP+.

For the proposed approach, we have used 300 atoms in the dictionary and let \( L = 20 \) for each iteration of G-SOMP+, which processes 8 × 8 image patches. We have chosen \( \eta = 10^{-9} \) in (6.7) and the residual decay parameter of G-SOMP+,
Table 6.1: Benchmarking on CAVE database [226]: The reported RMSE values are in the range of 8 bit images. The best results are shown in bold. The approaches highlighted in red also require the knowledge of spatial transform between the input images, which restrict their practical applicability.

<table>
<thead>
<tr>
<th>Method</th>
<th>Beads</th>
<th>Sponges</th>
<th>Spools</th>
<th>Painting</th>
<th>Pepper</th>
<th>Photos</th>
<th>Cloth</th>
<th>Statue</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSM [3]</td>
<td>28.5</td>
<td>19.9</td>
<td>-</td>
<td>12.2</td>
<td>13.7</td>
<td>13.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MF [109]</td>
<td>8.2</td>
<td>3.7</td>
<td>8.4</td>
<td>4.4</td>
<td>4.6</td>
<td>3.3</td>
<td>6.1</td>
<td>2.7</td>
</tr>
<tr>
<td>SASFM [91]</td>
<td>9.2</td>
<td>5.3</td>
<td>6.1</td>
<td>4.3</td>
<td>6.3</td>
<td>3.7</td>
<td>10.2</td>
<td>3.3</td>
</tr>
<tr>
<td>ADMM [215]</td>
<td>6.1</td>
<td>2.0</td>
<td>5.3</td>
<td>6.7</td>
<td>2.1</td>
<td>3.4</td>
<td>9.5</td>
<td>4.3</td>
</tr>
<tr>
<td>CMF [227]</td>
<td>6.6</td>
<td>4.0</td>
<td>15.0</td>
<td>26.0</td>
<td>5.5</td>
<td>11.0</td>
<td>20.0</td>
<td>16.0</td>
</tr>
<tr>
<td>Proposed</td>
<td>6.1</td>
<td>2.3</td>
<td>5.0</td>
<td>4.0</td>
<td>3.0</td>
<td>2.2</td>
<td>4.0</td>
<td>2.1</td>
</tr>
</tbody>
</table>

Table 6.2: Benchmarking on Harvard database [40]: The reported RMSE values are in the range of 8 bit images. The best results are shown in bold.

<table>
<thead>
<tr>
<th>Method</th>
<th>Img 1</th>
<th>Img b5</th>
<th>Img b8</th>
<th>Img d4</th>
<th>Img d7</th>
<th>Img h2</th>
<th>Img h3</th>
<th>Img f2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MF [109]</td>
<td>3.9</td>
<td>2.8</td>
<td>6.9</td>
<td>3.6</td>
<td>3.9</td>
<td>3.7</td>
<td>2.1</td>
<td>3.1</td>
</tr>
<tr>
<td>SASFM [91]</td>
<td>4.3</td>
<td>2.6</td>
<td>7.6</td>
<td>4.0</td>
<td>4.0</td>
<td>4.1</td>
<td>2.3</td>
<td>2.9</td>
</tr>
<tr>
<td>Proposed</td>
<td>1.2</td>
<td>0.9</td>
<td>5.9</td>
<td>2.4</td>
<td>2.1</td>
<td>1.0</td>
<td>0.5</td>
<td>1.7</td>
</tr>
</tbody>
</table>

\( \gamma = 0.99 \). We have optimized these parameter values, and the parameter settings of MF and SASFM, using a separate training set of 30 images. The training set comprises 15 images selected at random from each of the used databases. We defer further discussion on the parameter value selection for the proposed approach to Section 6.6.

Results on the images from the Harvard database [40] are shown in Table 6.2. In this table, we have compared the results of the proposed approach only with MF and SASFM because, like our approach, only these two approaches do not require the knowledge of the spatial transform between the input images. The table shows that the proposed approach consistently performs better than others. We have also experimented with the hyperspectral data that is remotely sensed by the NASA’s Airborne Visible and Infrared Imaging Spectrometer (AVIRIS) [80]. AVIRIS samples
6.5. Experimental Results

Figure 6.4: Spectral images for AVIRIS data at 460, 540, 620 and 1300 nm. Top row: 16 × 16 low spatial resolution hyperspectral (HS) image. Second row: 512 × 512 ground truth image. Third row: Estimated 512 × 512 HS image. Fourth row: Corresponding error image, with the scale is in the range of 8 bit images.

the scene reflectance in the wavelength range 400 - 2500 nm at a nominal interval of 10 nm. We have used a hyperspectral image taken over the Cuprite mines, Nevada\(^6\). The image has dimensions 512 × 512 × 224, where 224 represents the number of spectral bands in the image. Following [102], we have removed the bands 1-2, 105-115, 150-170 and 223-224 of the image because of extremely low SNR and water absorptions in those bands. We perform the down-sampling on the image as before and construct \(Y\) by directly selecting the 512 × 512 spectral images from the ground truth image, corresponding to the wavelengths 480, 560, 660, 830, 1650 and 2220 nm. These wavelengths correspond to the visible and mid-infrared range spectral channels of USGS/NASA Landsat 7 satellite\(^7\). We adopt this strategy of

\(^6\)Available at http://aviris.jpl.nasa.gov/data/free_data.html.
\(^7\)http://www.satimagingcorp.com/satellite-sensors/landsat.html.
Constructing $\mathbf{Y}$ from Huang et al. [91]. Fig. 6.4 shows the results of our approach for the estimation of the super-resolution hyperspectral image at 460, 540, 620 and 1300 nm. For this data set, the RMSE values for the proposed approach, MF [109] and SASFM [91] are 2.14, 3.06 and 3.11, respectively.

6.6 Discussion

G-SOMP+ uses two parameters. $L$: the number of dictionary atoms selected in each iteration, and $\gamma$: the residual decay parameter. By selecting more dictionary atoms in each iteration, G-SOMP+ computes the solution more quickly. The processing time of G-SOMP+ as a function of $L$, is shown in Fig. 6.5a. Each curve in Fig. 6.5 represents the mean values computed over a separate training data set of 15 images randomly selected from the database, whereas the dictionary used by G-SOMP+ contained 300 atoms. Fig. 6.5a shows the timings on an Intel Core i7-2600 CPU at 3.4 GHz with 8 GB RAM. Fig. 6.5b shows the RMSE values on the training data set as a function of $L$. Although, the error is fairly small over the complete range of $L$, the values are particularly low for $L \in \{15, ..., 25\}$, for both of the databases. Therefore, we have chosen $L = 20$ for all the test images in our experiments. Incidentally, the number of distinct spectral sources in a typical remote sensing hyperspectral image is also considered to be close to 20 [110]. Therefore, we have used the same value of the parameter for the remote sensing test image.
Figure 6.6: Selecting the number of dictionary atoms: RGB image of ‘Sponges’, containing roughly 7 – 10 distinct colors (materials), is shown on the left. Two dictionaries, with 10 and 50 atoms, are learned for the scene. After clustering the spectra (i.e. the dictionary atoms) into seven clusters (C1 - C7), it is visible that the dictionary with 50 atoms learns distinct clusters for the blue (C1) and the green (C2) sponges, whereas the dictionary with 10 atoms is not able to clearly distinguish between these sponges.

Generally, it is hard to know \( a \text{ priori} \) the exact number of iterations required by a greedy pursuit algorithm to converge. Similarly, if the residual error (i.e. \( ||R^t||_2 \) in Algorithm 7) is used as the stopping criterion, it is often difficult to select a single best value of this parameter for all the images. Fig. 6.5b shows that the RMSE curves rise for the higher values of \( L \) after touching a minimum value. In other words, more than the required number of dictionary atoms adversely affect the signal approximation. We use this observation to decide on the stopping criterion of G-SOMP+. Since the algorithm selects a constant number of atoms in each iteration, it stops if the approximation residual in its current iteration is more than a fraction \( \gamma \) of the residual in the previous iteration. As the approximation residual generally decreases rapidly before increasing (or becoming constant in some cases), we found that the performance of G-SOMP+ on the training images was mainly insensitive for \( \gamma \in [0.75, 1] \). From this range, we have selected \( \gamma = 0.99 \) for the test images in our experiments.

Our approach uses the online-dictionary learning technique [135] to solve (6.7). This technique needs to know the total number of dictionary atoms to be learned \( a \text{ priori} \). In Section 6.4, we have argued to use more dictionary atoms than the number of distinct materials in the scene. This results in a better separation of the spectral signal sources in the scene. Fig. 6.6 illustrates this notion. The figure shows an RGB image of ‘Sponges’ on the left. To extract the reflectance spectra, we learn two different dictionaries with 10 and 50 atoms, respectively, using the
16 × 16 hyperspectral image of the scene. We cluster the atoms of these dictionaries based on their correlation and show the arranged dictionaries in Fig. 6.6. From the figure, we can see that the dictionary with 10 atoms is not able to clearly distinguish between the reflectance spectra of the blue (C1) and the green (C2) sponge, whereas 10 seems to be a reasonable number representing the distinct materials in the scene. On the other hand, the dictionary with 50 atoms has learned two separate clusters for the two sponges.

The results reported in Fig. 6.5 are relatively insensitive to the number of dictionary atoms in the range of 50 to 300. In all our experiments, the proposed approach has learned a dictionary with 300 atoms. We choose a larger number to further incorporate the spectral variability of highly mixed scenes. Only in the image ‘Statue’ of Table 6.1, we used 150 dictionary atoms because of very low signal variability.

6.7 Conclusion

We have proposed a sparse representation based approach for hyperspectral image super-resolution. The proposed approach fuses a high spatial (but low spectral) resolution image with the hyperspectral image of the same scene. It uses the input low resolution hyperspectral image to learn a dictionary by solving a constrained sparse optimization problem. The atoms of the learned dictionary represent the reflectance spectra related to the scene. The learned dictionary is transformed according to the spectral quantization of the input high resolution image. This image and the transformed dictionary are later employed by an algorithm G-SOMP+. The proposed algorithm efficiently solves a constrained simultaneous sparse approximation problem to learn a sparse code. This sparse code is used with the originally learned dictionary to estimate the super-resolution hyperspectral image of the scene. We have tested our approach using the hyperspectral images of objects, real-world indoor and outdoor scenes and a remotely sensed hyperspectral image. Results of the experiments demonstrate that by taking advantage of the signal sparsity, non-negativity and the spatial structure in the scene, the proposed approach is generally able to perform better than the existing state of the art methods.
CHAPTER 7

Bayesian Sparse Representation for
Hyperspectral Image Super-Resolution

Abstract

Despite the proven efficacy of hyperspectral imaging in many computer vision tasks, its widespread use is hindered by its low spatial resolution, resulting from hardware limitations. We propose a hyperspectral image super resolution approach that fuses a high resolution image with the low resolution hyperspectral image using non-parametric Bayesian sparse representation. The proposed approach first infers probability distributions for the material spectra in the scene and their proportions. The distributions are then used to compute sparse codes of the high resolution image. To that end, we propose a generic Bayesian sparse coding strategy to be used with Bayesian dictionaries learned with the Beta process. We theoretically analyze the proposed strategy for its accurate performance. The computed codes are used with the estimated scene spectra to construct the super resolution hyperspectral image. Exhaustive experiments on two public databases of ground based hyperspectral images and a remotely sensed image show that the proposed approach outperforms the existing state of the art.

Keywords: Hyperspectral, super-resolution, Bayesian sparse representation, Beta Process, non-parametric sparse representation.

7.1 Introduction

Spectral characteristics of hyperspectral imaging have recently been reported to enhance performance in many computer vision tasks, including tracking [152], recognition and classification [73], [231], [201], segmentation [192] and document analysis [111]. They have also played a vital role in medical imaging [241], [109] and remote sensing [24], [9]. Hyperspectral imaging acquires a faithful spectral representation of the scene by integrating its radiance against several spectrally well-localized basis functions. However, contemporary hyperspectral systems lack in spatial resolution [8], [109], [47]. This fact is impeding their widespread use. In this regard, a simple solution of using high resolution sensors is not viable as it

---

Published in Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2015.
further reduces the density of the photons reaching the sensors, which is already limited by the high spectral resolution of the instruments.

Due to hardware limitations, software based approaches for hyperspectral image super resolution (e.g. see Fig. 7.1) are considered highly attractive [8]. At present, the spatial resolution of the systems acquiring images by a gross quantization of the scene radiance (e.g. RGB and RGB-NIR) is much higher than their hyperspectral counterparts. In this work, we propose to fuse the spatial information from the images acquired by these systems with the hyperspectral images of the same scenes using non-parametric Bayesian sparse representation.

The proposed approach fuses a hyperspectral image with the high resolution image in a four-stage process, as shown in Fig. 7.2. In the first stage, it infers probability distributions for the material reflectance spectra in the scene and a set of Bernoulli distributions, indicating their proportions in the image. Then, it estimates a dictionary and transforms it according to the spectral quantization of the high resolution image. In the third stage, the transformed dictionary and the Bernoulli distributions are used to compute the sparse codes of the high resolution image. To that end, we propose a generic Bayesian sparse coding strategy to be used with Bayesian dictionaries learned with the Beta process [157]. We theoretically analyze the proposed strategy for its accurate performance. Finally, the computed codes are used with the estimated dictionary to construct the super resolution hyperspectral image. The proposed approach not only improves the state of the art results, which is verified by exhaustive experiments on three different public data sets, it also maintains the advantages of the non-parametric Bayesian framework over the typical optimization based approaches [8], [109], [215], [227].

The rest of the paper is organized as follows. After reviewing the related literature in Section 7.2, we formalize the problem in Section 7.3. The proposed approach
Figure 7.2: Schematics of the proposed approach: (1) Sets of distributions over the dictionary atoms and the support indicator vectors are inferred non-parametrically. (2) A dictionary $\Phi$ is estimated and transformed according to the spectral quantization of the high resolution image $Y$. (3) The transformed dictionary and the distributions over the support indicator vectors are used for sparse coding $Y$. This step is performed by the proposed Bayesian sparse coding strategy. (4) The codes are used with $\Phi$ to construct the target super resolution image.

is presented in Section 7.4 and evaluated in Section 7.5. Section 7.6 provides a discussion on the parameter settings, and Section 7.7 concludes the paper.

7.2 Related Work

Hyperspectral sensors have been in use for nearly two decades in remote sensing [24]. However, it is still difficult to obtain high resolution hyperspectral images by the satellite sensors due to technical and budget constraints [91]. This fact has motivated considerable research in hyperspectral image super resolution, especially for remote sensing. To enhance the spatial resolution, hyperspectral images are usually fused with the high resolution pan-chromatic images (i.e. pan-sharpening) [192], [47]. In this regard, conventional approaches are generally based on projection and substitution, including the intensity hue saturation [88] and the principle component analysis [46]. In [3] and [37], the authors have exploited the sensitivity of human vision to luminance and fused the luminance component of the high resolution im-
ages with the hyperspectral images. However, this approach can also cause spectral distortions in the resulting image [39].

Minghelli-Roman et al. [144] and Zhukov et al. [243] have used hyperspectral unmixing [110], [6] for spatial resolution enhancement of hyperspectral images. However, their methods require that the spectral resolutions of the images being fused are close to each other. Furthermore, these approaches struggle in highly mixed scenarios [91]. Zurita-Milla et al. [244] have enhanced their performance for such cases using the sliding window strategy.

More recently, matrix factorization based hyperspectral image super resolution for ground based and remote sensing imagery has been actively investigated [109], [215], [91], [227], [8]. Approaches developed under this framework fuse high resolution RGB images with hyperspectral images. Kawakami et al. [109] represented each image from the two modalities by two factors and constructed the desired image with the complementary factors of the two representations. Similar approach is applied in [91] to the remotely acquired images, where the authors used a down-sampled version of the RGB image in the fusion process. Wycoff et al. [215] developed a method based on Alternating Direction Method of Multipliers (ADMM) [30]. Their approach also requires prior knowledge about the spatial transform between the images being fused. Akhtar et al. [8] proposed a method based on sparse spatio-spectral representation of hyperspectral images that also incorporates the non-negativity of the spectral signals. The strength of the approach comes from exploiting the spatial structure in the scene, which requires processing the images in terms of spatial patches and solving a simultaneous sparse optimization problem [199]. Yokoya et al. [227] made use of coupled feature space between a hyperspectral and a multispectral image of the same scene.

Matrix factorization based approaches have been able to show state of the art results in hyperspectral image super resolution using the image fusion technique. However, Akhtar et al. [8] showed that their performance is sensitive to the algorithm parameters, especially to the sizes of the matrices (e.g. dictionary) into which the images are factored. Furthermore, there is no principled way to incorporate prior domain knowledge to enhance the performance of these approaches.

### 7.3 Problem Formulation

Let $Y^h \in \mathbb{R}^{m \times n \times L}$ be the acquired low resolution hyperspectral image, where $L$ denotes the spectral dimension. We assume availability of a high resolution image $Y \in \mathbb{R}^{M \times N \times l}$ (e.g. RGB) of the same scene, such that $M \gg m$, $N \gg n$ and $L \gg l$. 
7.4. Proposed Approach

Our objective is to estimate the super resolution hyperspectral image \( T \in \mathbb{R}^{M \times N \times L} \) by fusing \( Y \) and \( Y^h \). For our problem, \( Y^h = \Psi^h(T) \) and \( Y = \Psi(T) \), where \( \Psi^h : \mathbb{R}^{M \times N \times L} \to \mathbb{R}^{m \times n \times L} \) and \( \Psi : \mathbb{R}^{M \times N \times L} \to \mathbb{R}^{M \times N \times l} \).

Let \( \Phi \in \mathbb{R}^{L \times |K|} \) be an unknown matrix with columns \( \varphi_k \), where \( k \in \mathcal{K} = \{1, ..., K\} \) and \( |.| \) denotes the cardinality of the set. Let \( Y^h = \Phi B \), where the matrix \( Y^h \in \mathbb{R}^{L \times mn} \) is created by arranging the pixels of \( Y^h \) as its columns and \( B \in \mathbb{R}^{|K| \times mn} \) is a coefficient matrix. For our problem, the basis vectors \( \varphi_k \) represent the reflectance spectra of different materials in the imaged scene. Thus, we also allow for the possibility that \( |\mathcal{K}| > L \). Normally, \( |\mathcal{K}| \ll mn \) because a scene generally comprises only a few spectrally distinct materials [8]. Let \( \hat{\Phi} \in \mathbb{R}^{L \times |K|} \) be such that \( Y = \hat{\Phi} A \), where \( Y \in \mathbb{R}^{L \times MN} \) is formed by arranging the pixels of \( Y \) and \( A \in \mathbb{R}^{|K| \times MN} \) is a coefficient matrix. The columns of \( \hat{\Phi} \) are also indexed in \( \mathcal{K} \). Since \( Y^h \) and \( Y \) represent the images of the same scene, \( \hat{\Phi} = \Upsilon \Phi \), where \( \Upsilon \in \mathbb{R}^{L \times L} \) is a transformation matrix, associating the spectral quantizations of the two imaging modalities. Similar to the previous works [8], [109], [215], this transform is considered to be known \textit{a priori}.

In the above formulation, pixels of \( Y \) and \( Y^h \) are likely to admit sparse representations over \( \hat{\Phi} \) and \( \Phi \), respectively, because a pixel generally contains very few spectra as compared to the whole image. Furthermore, the value of \( |\mathcal{K}| \) can vary greatly between different scenes, depending on the number of spectrally distinct materials present in a scene. In the following, we refer to \( \Phi \) as the \textit{dictionary} and \( \hat{\Phi} \) as the \textit{transformed dictionary}. The columns of the dictionaries are called their \textit{atoms} and a complementary coefficient matrix (e.g. \( A \)) is referred as the \textit{sparse code matrix} or the \textit{sparse codes} of the corresponding image. We adopt these conventions from the sparse representation literature [176].

7.4 Proposed Approach

We propose a four-stage approach for hyperspectral image super resolution that is illustrated in Fig. 7.2. The proposed approach first separates the scene spectra by learning a dictionary from the low resolution hyperspectral image under a Bayesian framework. The dictionary is transformed using the known spectral transform \( \Upsilon \) between the two input images as \( \hat{\Phi} = \Upsilon \Phi \). The transformed dictionary is used for encoding the high-resolution image. The codes \( \tilde{A} \in \mathbb{R}^{|K| \times MN} \) are computed using the proposed strategy. As shown in the figure, we eventually use the dictionary and the codes to construct \( T = \Phi \tilde{A} \), where \( T \in \mathbb{R}^{L \times MN} \) is formed by arranging the pixels of the target image \( T \). Hence, accurate estimation of \( \tilde{A} \) and \( \Phi \) is crucial
for our approach, where the dictionary estimation also includes finding its correct size, i.e. $|\mathcal{K}|$. Furthermore, we wish to incorporate the ability of using the prior domain knowledge in our approach. This naturally leads towards exploiting the non-parametric Bayesian framework. The proposed approach is explained below, following the sequence in Fig. 7.2.

### 7.4.1 Bayesian Dictionary Learning

We denote the $i$th pixel of $Y^h$ by $y^h_i \in \mathbb{R}^L$, that admits to a sparse representation $\beta^h_i \in \mathbb{R}^{|\mathcal{K}|}$ over the dictionary $\Phi$ with a small error $\epsilon^h_i \in \mathbb{R}^L$. Mathematically, $y^h_i = \Phi \beta^h_i + \epsilon^h_i$. To learn the dictionary in these settings\(^1\), Zhou et al. [238] proposed a beta process [157] based non-parametric Bayesian model, that is shown below in its general form. In the given equations and the following text, we have dropped the superscript ‘$h$’ for brevity, as it can be easily deduced from the context.

- $y_i = \Phi \beta_i + \epsilon_i \quad \forall i \in \{1, ..., mn\}$
- $\beta_i = z_i \odot s_i$
- $\varphi_k \sim \mathcal{N}(\varphi_k | \mu_{\kappa_o}, \Lambda_{\kappa_o}^{-1}) \quad \forall k \in \mathcal{K}$
- $z_{ik} \sim \text{Bern}(z_{ik} | \pi_{\kappa_o})$
- $\pi_k \sim \text{Beta}(\pi_k | a_o/K, b_o(K - 1)/K)$
- $s_{ik} \sim \mathcal{N}(s_{ik} | \mu_{s_o}, \lambda_{s_o}^{-1})$
- $\epsilon_i \sim \mathcal{N}(\epsilon_i | 0, \Lambda_{\kappa_o}^{-1})$.

In the above model, $\odot$ denotes the Hadamard/element-wise product; $\sim$ denotes a draw (i.i.d.) from a distribution; $\mathcal{N}$ refers to a Normal distribution; Bern and Beta represent Bernoulli and Beta distributions, respectively. Furthermore, $z_i \in \mathbb{R}^{|\mathcal{K}|}$ is a binary vector whose $k^{th}$ component $z_{ik}$ is drawn from a Bernoulli distribution with parameter $\pi_{\kappa_o}$. Conjugate Beta prior is placed over $\pi_k$, with hyper-parameters $a_o$ and $b_o$. We have used the subscript ‘$o$’ to distinguish the parameters of the prior distributions. We refer to $z_i$ as the support indicator vector, as the value $z_{ik} = 1$ indicates that the $k^{th}$ dictionary atom participates in the expansion of $y_i$. Also, each component $s_{ik}$ of $s_i \in \mathbb{R}^{|\mathcal{K}|}$ (the weight vector) is drawn from a Normal distribution.

For tractability, we restrict the precision matrix $\Lambda_{\kappa_o}$ of the prior distribution over a dictionary atom to $\lambda_{\kappa_o} I_L$, where $I_L$ denotes the identity in $\mathbb{R}^{L \times L}$ and $\lambda_{\kappa_o} \in \mathbb{R}$ is a predetermined constant. A zero vector is used for the mean parameter $\mu_{\kappa_o} \in \mathbb{R}^L$.

\(^1\)The sparse code matrix $B$ (with $\beta_i \in \{1, ..., mn\}$ as its columns) is also learned. However, it is not required by our approach.
7.4. Proposed Approach

since the distribution is defined over a basis vector. Similarly, we let \( \Lambda_c = \lambda_c I_L \)
and \( \mu_{so} = 0 \), where \( \lambda_c \in \mathbb{R} \). These simplifications allow for fast inferencing in
our application without any noticeable degradation of the results. We further place non-informative gamma hyper-priors over \( \lambda_{so} \) and \( \lambda_{co} \), so that \( \lambda_s \sim \Gamma(\lambda_s|c_o, d_o) \) and \( \lambda_c \sim \Gamma(\lambda_c|e_o, f_o) \), where \( \Gamma \) denotes the Gamma distribution and \( c_o, d_o, e_o \) and \( f_o \) are
the hyper-parameters. The model thus formed is completely conjugate, therefore Bayesian inferencing can be performed over it with Gibbs sampling using analytical expressions. We derive these expressions for the proposed approach and state the
final sampling equations below. Detailed derivations of the Gibbs sampling equations
can be found in the Appendix B.1.

We denote the contribution of the \( k \)-th dictionary atom \( \varphi_k \) to \( y_i \) as,
\[ y_i \varphi_k = y_i - \Phi(z_i \odot s_i) + \varphi_k(z_{ik} s_{ik}) , \]
and the \( \ell_2 \) norm of a vector by \( \| \cdot \|_2 \). Using these
notations, we obtain the following analytical expressions for the Gibbs sampling
process used in our approach:

**Sample** \( \varphi_k \): from \( \mathcal{N}(\varphi_k|\mu_k, \lambda_k^{-1} I_L) \), where
\[ \lambda_k = \lambda_{ko} + \lambda_{co} \sum_{i=1}^{mn} (z_{ik} s_{ik})^2 ; \mu_k = \frac{\lambda_{ko}}{\lambda_k} \sum_{i=1}^{mn} (z_{ik} s_{ik}) y_i \varphi_k . \]

**Sample** \( z_{ik} \): from Bern \( \left( z_{ik} | 1 - \frac{\xi_{ko}}{1 - \xi_{ko} + \xi \mu_{ko}} \right) \), where
\[ \xi = \exp \left( - \frac{\lambda_{ko}}{2} (\varphi_k^T \varphi_k s_{ik}^2 - 2 s_{ik} y_i \varphi_k) \right) . \]

**Sample** \( s_{ik} \): from \( \mathcal{N}(s_{ik}|\mu_s, \lambda_s^{-1}) \), where
\[ \lambda_s = \lambda_{so} + \lambda_{co} z_{ik}^2 \varphi_k \varphi_k^T ; \mu_s = \frac{\lambda_{so}}{\lambda_s} s_{ik} \varphi_k y_i \varphi_k . \]

**Sample** \( \pi_k \): from Beta \( (\pi_k | a, b) \), where
\[ a = \frac{a_o}{K} + \sum_{i=1}^{mn} z_{ik} ; b = \frac{b_o (K - 1)}{K} + (mn - \sum_{i=1}^{mn} z_{ik}) . \]

**Sample** \( \lambda_s \): from \( \Gamma(\lambda_s|c, d) \), where
\[ c = \frac{K mn}{2} + c_o ; d = \frac{1}{2} \sum_{i=1}^{mn} ||s_i||_2^2 + d_o . \]

**Sample** \( \lambda_c \): from \( \Gamma(\lambda_c|e, f) \), where
\[ e = \frac{L mn}{2} + e_o ; f = \frac{1}{2} \sum_{i=1}^{mn} ||y_i - \Phi(z_i \odot s_i)||_2^2 + f_o . \]
As a result of Bayesian inferencing, we obtain sets of posterior distributions over the model parameters. We are interested in two of them. (a) The set of distributions over the atoms of the dictionary, \( \mathcal{R} = \{ \mathcal{N}(\varphi_k|\mu_k, \Lambda_k^{-1}) : k \in \mathcal{K} \} \subset \mathbb{R}^L \) and (b) the set of distributions over the components of the support indicator vectors \( \mathcal{S} = \{ \text{Bern}(\pi_k) : k \in \mathcal{K} \} \subset \mathbb{R} \). Here, \( \text{Bern}(\pi_k) \) is followed by the \( k^{th} \) components of all the support indicator vectors simultaneously, i.e. \( \forall i \in \{1, ..., mn\}, z_{ik} \sim \text{Bern}(\pi_k) \). These sets are used in the later stages of the proposed approach.

In the above model, we have placed Gaussian priors over the dictionary atoms, enforcing our prior belief of relative smoothness of the material spectra. Note that, the correct value of \( |\mathcal{K}| \) is also inferred at this stage. We refer to the pioneering work by Paisley and Carin [157] for the theoretical details in this regard. In our inferencing process, the desired value of \( |\mathcal{K}| \) manifests itself as the total number of dictionary atoms for which \( \pi_k \neq 0 \) after convergence. To implement this, we start with \( K \to \infty \) and later drop the dictionary atoms corresponding to \( \pi_k = 0 \) during the sampling process.

With the computed \( \mathcal{R} \), we estimate \( \Phi \) (stage 2 in Fig. 7.2) by drawing multiple samples from the distributions in the set and computing their respective means. It is also possible to directly use the mean parameters of the inferred distributions as the estimates of the dictionary atoms, but the former is preferred for robustness. Henceforth, we will consider the dictionary, instead of the distributions over its atoms, as the final outcome of the Bayesian dictionary learning process. The transformed dictionary is simply computed as \( \hat{\Phi} = \Upsilon \Phi \). Recall that, the matrix \( \Upsilon \) relates the spectral quantizations of the two imaging modalities under consideration and it is known \textit{a priori}.

### 7.4.2 Bayesian Sparse Coding

Once \( \hat{\Phi} \) is known, we use it to compute the sparse codes of \( Y \). The intention is to obtain the codes of the high resolution image and use them with \( \Phi \) to estimate \( T \). Although some popular strategies for sparse coding already exist, e.g. Orthogonal Matching Pursuit [197] and Basis Pursuit [48], but their performance is inferior when used with the Bayesian dictionaries learned using the Beta process. There are two main reasons for that. (a) Atoms of the Bayesian dictionaries are not constrained to \( \ell_2 \) unit norm. (b) With these atoms, there is an associated set of Bernoulli distributions which must not be contradicted by the underlying support of the sparse codes. In some cases, it may be easy to modify an existing strategy to
7.4. Proposed Approach

cater for (a), but it is not straightforward to take care of (b) in these approaches.

We propose a simple, yet effective method for Bayesian sparse coding that can be
generically used with the dictionaries learned using the Beta process. The proposal
is to follow a procedure similar to the Bayesian dictionary learning, with three major
differences. For a clear understanding, we explain these differences as modifications
to the inferencing process of the Bayesian dictionary learning, following the same
notational conventions as above.

1) Use $N(\hat{\varphi}_k|\mu_{k_o},\lambda^{-1}_kI_l)$ as the prior distribution over the $k^{th}$ dictionary atom,
where $\lambda_{k_o} \rightarrow \infty$ and $\mu_{k_o} = \hat{\varphi}_k$. Considering that $\hat{\Phi}$ is already a good estimate of the
dictionary\(^2\), this is an intuitive prior. It entails, $\hat{\varphi}_k$ is sampled from the following
posterior distribution while inferencing:
Sample $\hat{\varphi}_k$: from $N(\hat{\varphi}_k|\mu_k,\lambda^{-1}_kI_l)$, where
\[
\lambda_k = \lambda_{k_o} + \lambda_{\epsilon_o} \sum_{i=1}^{MN} (z_{ik}s_{ik})^2;
\]
\[
\mu_k = \frac{\lambda_{\epsilon_o}}{\lambda_k} \sum_{i=1}^{MN} (z_{ik}s_{ik})_{i \neq k} + \frac{\lambda_{k_o}}{\lambda_k} \mu_{k_o}.
\]
In the above equations, $\lambda_{k_o} \rightarrow \infty$ signifies $\lambda_k \approx \lambda_{k_o}$ and $\mu_k \approx \mu_{k_o}$.
It further implies that we are likely to get similar samples against multiple draws from the distribution.
In other words, we can not only ignore to update the posterior distributions over the
dictionary atoms during the inferencing process, but also approximate them with a
fixed matrix. A sample from the $k^{th}$ posterior distribution is then the
$k^{th}$ column of this matrix. Hence, from the implementation perspective, Bayesian sparse coding
directly uses the atoms of $\hat{\Phi}$ as the samples from the posterior distributions.

2) Sample the support indicator vectors in accordance with the Bernoulli dis-
tributions associated with the fixed dictionary atoms. To implement this, while
inferencing, we fix the distributions over the support indicator vectors according
to $\mathcal{S}$. As shown in Fig. 7.2, we use the vector $\pi \in \mathbb{R}^{|K|}$ for this purpose, which
stores the parameters of the distributions in the set $\mathcal{S}$. While sampling, we directly
use the $k^{th}$ component of $\pi$ as $\pi_k$. It is noteworthy that using $\pi$ in coding $Y$
also imposes the self-consistency of the scene between the high resolution image $Y$ and
the hyperspectral image $Y^h$.

Incorporating the above proposals in the Gibbs sampling process and performing
the inferencing can already result in a reasonably accurate sparse representation of

\(^2\)This is true because $\hat{\Phi}$ is an exact transform of $\Phi$, which is computed with high confidence.
Chapter 7. Bayesian Sparse Representation for HSI Super-Resolution

Indeed, a closer examination of the underlying probabilistic settings reveals that a more accurate estimate of the sparse codes is readily obtainable.

**Lemma 1.** With $y \in \mathcal{R}(\hat{\Phi})$ (i.e. $\exists \alpha$ s.t. $y = \hat{\Phi}\alpha$) and $|K| > l$, the best estimate of the representation of $y$, in the mean squared error sense$^3$, is given by $\hat{\alpha}_{\text{opt}} = E[E[\alpha|z]]$, where $\mathcal{R}(\cdot)$ is the range operator, $E[\cdot]$ and $E[\cdot|\cdot]$ are the expectation and the conditional expectation operators, respectively.

**Proof:** Let $\hat{\alpha} \in \mathbb{R}^{|\hat{K}|}$ be an estimate of the representation $\alpha$ of $y$, over $\hat{\Phi}$. We can define the mean square error (MSE) as the following:

$$\text{MSE} = E[||\hat{\alpha} - \alpha||_2^2].$$

In our settings, the components of a support indicator vector $z$ are independent draws from Bernoulli distributions. Let $Z$ be the set of all possible support indicator vectors in $\mathbb{R}^{|K|}$, i.e. $|Z| = 2^{|K|}$. Thus, there is a non-negative probability of selection $P(z)$ associated with each $z \in Z$ such that $\sum_{z \in Z} P(z) = 1$. Indeed, the probability mass function $p(z)$ depends on the vector $\pi$ that assigns higher probabilities to the vectors indexing more important dictionary atoms.

We can model the generation of $\alpha$ as a two step sequential process: 1) Random selection of $z$ with probability $P(z)$. 2) Random selection of $\alpha$ according to a conditional probability density function $p(\alpha|z)$. Here, the selection of $\alpha$ implies the selection of the corresponding weight vector $s$ and then computing $\alpha = z \odot s$. Under this perspective, MSE can be re-written as:

$$\text{MSE} = \sum_{z \in Z} P(z) E[||\hat{\alpha} - \alpha||_2^2 | z].$$

The conditional expectation in (7.2) can be written as:

$$E[||\hat{\alpha} - \alpha||_2^2 | z] = ||\hat{\alpha}\|^2_2 - 2\alpha^T E[\alpha|z] + E[||\alpha||_2^2|z].$$

We can write the last term in (7.3) as the following:

$$E[||\alpha||_2^2|z] = ||E[\alpha|z]\|^2_2 + E[||\alpha - E[\alpha|z]\||_2^2|z].$$

For brevity, let us denote the second term in (7.4) as $V_z$. By combining (7.2)-(7.4) we get the following:

$$\text{MSE} = \sum_{z \in Z} P(z) ||\hat{\alpha} - E[\alpha|z]\||_2^2 + \sum_{z \in Z} P(z)V_z$$

$$= E[||\hat{\alpha} - E[\alpha|z]\||_2^2] + E[V_z].$$

---

$^3$The metric is chosen based on the existing literature [109],[91],[8].
Differentiating R.H.S. of (7.6) with respect to $\tilde{\alpha}$ and equating it to zero, we get $\tilde{\alpha}_{\text{opt}} = \mathbb{E}[\mathbb{E}[\alpha|z]]^4$, that minimizes the mean squared error.

Notice that, with the aforementioned proposals incorporated in the sampling process, it is possible to independently perform the inferencing multiple, say $Q$, times. This would result in $Q$ support indicator vectors $z_q$ and weight vectors $s_q$ for $y$, where $q \in \{1,...Q\}$.

Lemma 2. For $Q \to \infty$, $\frac{1}{Q} \sum_{q=1}^{Q} z_q \odot s_q = \mathbb{E}[\mathbb{E}[\alpha|z]]$.

**Proof:** We only discuss an informal proof of Lemma 2. The following statements are valid in our settings:

(a) $\exists \alpha_i, \alpha_j \text{ s.t. } (\alpha_i \neq \alpha_j) \land (\alpha_i = z \odot s_i) \land (\alpha_j = z \odot s_j)$,
(b) $\exists z_i, z_j \text{ s.t. } (z_i \neq z_j) \land (\alpha = z_i \odot s_i) \land (\alpha = z_j \odot s_j)$,

where $\land$ denotes the logical and; $\alpha_i$ and $\alpha_j$ are instances of two distinct solutions of the underdetermined system $y = \hat{\Phi}\alpha$. In the above statements, (a) refers to the possibility of distinct representations with the same support and (b) refers to the existence of distinct support indicator vectors for a single representation. Validity of these conditions can be easily verified by noticing that $z$ and $s$ are allowed to have zero components. For a given inferencing process, the final computed vectors $z$ and $s$ are drawn according to valid probability distributions. Thus, (a) and (b) entail that the mean of $Q$ independently computed representations, is equivalent to $\mathbb{E}[\mathbb{E}[\alpha|z]]$ when $Q \to \infty$.

3) In the light of Lemma 1 and 2, we propose to independently repeat the inferencing process $Q$ times, where $Q$ is a large number (e.g. 100), and finally compute the code matrix $\tilde{A}$ (in Fig. 7.2) as $\tilde{A} = \frac{1}{Q} \sum_{q=1}^{Q} Z_q \odot S_q$, where $\tilde{A}$ has $\tilde{\alpha}\in\{1,...,MN\}$ as its columns. The matrices $Z_q, S_q \in \mathbb{R}^{[K] \times MN}$ are the support matrix and the weight matrix, respectively, formed by arranging the support indicator vectors and the weight vectors as their columns. Note that, the finally computed codes $\tilde{A}$ may by dense as compared to individual $Z_q$.

With the estimated $\tilde{A}$ and the dictionary $\Phi$, we compute the target super resolution image $T$ by re-arranging the columns of $\tilde{T} = \Phi\tilde{A}$ (stage 4 in Fig. 7.2) into the pixels of hyperspectral image.

---

4Detailed derivation of each step used in the proof is also provided in Appendix B.2.
7.5 Experimental Evaluation

The proposed approach has been thoroughly evaluated using ground based imagery as well as remotely sensed data. For the former, we performed exhaustive experiments on two public databases, namely, the CAVE database [226] and the Harvard database [40]. CAVE comprises 32 hyperspectral images of everyday objects with dimensions $512 \times 512 \times 31$, where 31 represents the spectral dimension. The spectral images are in the wavelength range 400 - 700nm, sampled at a regular interval of 10nm. The Harvard database consists of 50 images of indoor and outdoor scenes with dimensions $1392 \times 1040 \times 31$. The spectral samples are taken at every 10nm in the range 420 - 720nm. For the remote sensing data, we chose a $512 \times 512 \times 224$ hyperspectral image\(^5\) acquired by the NASA’s Airborne Visible Infrared Imaging Spectrometer (AVIRIS) [80]. This image has been acquired over the Cuprite mines in Nevada, in the wavelength range 400 - 2500nm with 10nm sampling interval. We followed the experimental protocol of [8] and [109]. For benchmarking, we compared the results with the existing best reported results in the literature under the same protocol, unless the code was made public by the authors. In the latter case, we performed experiments using the provided code and the optimized parameter values. The reported results are in the range of 8 bit images.

In our experiments, we consider the images from the databases as the ground truth. A low resolution hyperspectral image $Y^h$ is created by averaging the ground truth over $32 \times 32$ spatially disjoint blocks. For the Harvard database, $1024 \times 1024 \times 31$ image patches were cropped from the top left corner of the images, to make the spatial dimensions of the ground truth multiples of 32. For the ground based imagery, we assume the high resolution image $Y$ to be an RGB image of the same scene. We simulate this image by integrating the ground truth over its spectral dimension using the spectral response of Nikon D700\(^6\). For the remote sensing data, we consider $Y$ to be a multispectral image. Following [8], we create this image by directly selecting six spectral images from the ground truth against the wavelengths 480, 560, 660, 830, 1650 and 2220 nm. Thus, in this case, $\Upsilon$ is a $6 \times 224$ binary matrix that selects the corresponding rows of $\Phi$. The mentioned wavelengths correspond to the visible and mid-infrared channels of USGS/NASA Landsat 7 satellite.

We compare our results with the recently proposed approaches, namely, the Matrix Factorization based method (MF) [109], the Spatial Spectral Fusion Model\(^7\).

\(^5\)http://aviris.jpl.nasa.gov/data/free_data.html
\(^6\)The response and integration limits can be found at http://www.maxmax.com/spectral_response.htm
### 7.5. Experimental Evaluation

Table 7.1: Benchmarking of the proposed approach: The RMSE values are in the range of 8 bit images. The best results are shown in bold. The approaches highlighted in red additionally require the knowledge of the spatial transform between the input images.

<table>
<thead>
<tr>
<th>Method</th>
<th>CAVE database [226]</th>
<th>Harvard database [40]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Beads</td>
<td>Spools</td>
</tr>
<tr>
<td>CSM [3]</td>
<td>28.5</td>
<td>-</td>
</tr>
<tr>
<td>MF [109]</td>
<td>8.2</td>
<td>8.4</td>
</tr>
<tr>
<td>SSFM [91]</td>
<td>9.2</td>
<td>6.1</td>
</tr>
<tr>
<td>ADMM [215]</td>
<td>6.1</td>
<td>5.3</td>
</tr>
<tr>
<td>CMF [227]</td>
<td>6.6</td>
<td>15.0</td>
</tr>
<tr>
<td>GSOMP [8]</td>
<td>6.1</td>
<td>5.0</td>
</tr>
<tr>
<td>Proposed</td>
<td><strong>5.4</strong></td>
<td><strong>4.6</strong></td>
</tr>
<tr>
<td></td>
<td>Img 1</td>
<td>Img b5</td>
</tr>
<tr>
<td>MF [109]</td>
<td>3.9</td>
<td>2.8</td>
</tr>
<tr>
<td>SSFM [91]</td>
<td>4.3</td>
<td>2.6</td>
</tr>
<tr>
<td>GSOMP [8]</td>
<td>1.2</td>
<td><strong>0.9</strong></td>
</tr>
<tr>
<td>Proposed</td>
<td><strong>1.1</strong></td>
<td><strong>0.9</strong></td>
</tr>
</tbody>
</table>

(SSFM) [91], the ADMM based approach [215], the Coupled Matrix Factorization method (CMF) [227] and the spatio-spectral sparse representation approach, GSOMP [8]. These matrix factorization based approaches constitute the state of the art in this area [8]. In order to show the performance difference between these methods and the other approaches mentioned in Section 7.2, we also report some results of the Component Substitution Method (CSM) [3], taken directly from [109].

The top half of Table 7.1 shows results on seven different images from the CAVE database. We chose these images because they are commonly used for benchmarking in the existing literature [8],[215],[109]. The table shows the root mean squared error (RMSE) of the reconstructed super resolution images. The approaches highlighted in red additionally require the knowledge of the down-sampling matrix that converts the ground truth to the acquired hyperspectral image. Hence, they are of less practical value [8]. As can be seen, our approach outperforms most of the existing methods by a considerable margin on all the images. Only the results of GSOMP are
Table 7.2: Exhaustive experiment results: The means and the standard deviations of the RMSE values are computed over the complete databases.

<table>
<thead>
<tr>
<th>Method</th>
<th>CAVE database [226]</th>
<th>Harvard database [40]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean ± Std. Dev</td>
<td>Mean ± Std. Dev</td>
</tr>
<tr>
<td>GSOMP [8]</td>
<td>3.66 ± 1.51</td>
<td>2.84 ± 2.24</td>
</tr>
<tr>
<td>Proposed</td>
<td>3.06 ± 1.12</td>
<td>1.74 ± 1.49</td>
</tr>
</tbody>
</table>

Figure 7.3: Comparison of the proposed approach with GSOMP [8] on image ‘Spools’ (CAVE database) [226].

comparable to our method. However, GSOMP operates under the assumption that nearby pixels in the target image are spectrally similar. The assumption is enforced with the help of two extra algorithm parameters. Fine tuning these parameters is often non-trivial, as many of the nearby pixels in an image can also have dissimilar spectra. There is no provision for automatic adjustment of the parameter values for such cases. Therefore, an image reconstructed by GSOMP can often suffer from spatial artifacts. For instance, even though the parameters of GSOMP are optimized specifically for the sample image in Fig. 7.3, the spatial artifacts are still visible. The figure also compares the RMSE of our approach with that of GSOMP, as a function of the spectral bands of the image. The RMSE curve is lower and smoother for the proposed approach.

The results on the images from the Harvard database are shown in the bottom half of Table 7.1. These results also favor our approach. Results of ADMM and CMF have never been reported for the Harvard database. In Table 7.2, we report the means and the standard deviations of the RMSE values of the proposed approach.
over the complete databases. The results are compared with GSOMP using the public code provided by the authors and the optimal parameter settings for each database, as mentioned in [8]. Fair comparison with other approaches is not possible because of the unavailability of the public codes and results on the full databases. However, based on Table 7.1 and the mean RMSE values of 4.24±0.08 and 4.98±1.97 for ADMM and MF, respectively, reported by Wycoff et al. [215], on 20 images from the CAVE database, we can safely conjecture that the other methods are unlikely to outperform our approach on the full databases. Table 7.2 clearly indicates the consistent performance of the proposed approach.

For qualitative analysis, Fig. 7.4 shows spectral samples from two reconstructed super resolution hyperspectral images, against wavelengths 460, 540 and 620nm. The spectral images are shown along the ground truth and their absolute difference with the ground truth. Spectral samples of the input 16 × 16 hyperspectral images are also shown. Successful hyperspectral image super resolution is clearly evident from the images. For the remote sensing image acquired by AVIRIS, the RMSE value of the proposed approach is 1.63. This is also lower than the previously reported values of 2.14, 3.06 and 3.11 for GSOMP, MF and SSFM respectively, in [8]. For the AVIRIS image, the spectral samples of the reconstructed super resolution image at 460, 540, 620 and 1300 nm are shown in Fig. 7.5.
Figure 7.5: Spectral images at 460, 540, 620 and 1300 nm for the AVIRIS [80] data. Reconstructed spectral images (512 × 512) are shown along their absolute difference with the ground truth (512 × 512). The corresponding low resolution images (16 × 16) are also shown.

7.6 Discussion

In the above experiments, we initialized the Bayesian dictionary learning stage as follows. The parameters $a_o, b_o, c_o, d_o, e_o$ and $f_o$ were set to $10^{-6}$. From the sampling equations in Section 7.4.1, it is easy to see that these values do not influence the posterior distributions much, and other such small values would yield similar results. We initialized $\pi_{k_o} = 0.5, \forall k$, to give the initial Bernoulli distributions the largest variance [26]. We initialized the Gibbs sampling process with $K = 50$ for all the images. This value is based on our prior belief that the total number of the materials in a given scene is generally less than 50. The final value of $|K|$ was inferred by the learning process itself, which ranged over 10 to 33 for different images. We initialized
$\lambda_{c_o}$ to the precision of the pixels in $Y^h$ and randomly chose $\lambda_{s_o} = 1$. Following [238], $\lambda_{k_o}$ was set to $L$. The parameter setting was kept unchanged for all the datasets without further fine tuning. We ran fifty thousand Gibbs sampling iterations from which the last 100 were used to sample the distributions to compute $\Phi$. On average, this process took around 3 minutes for the CAVE images and around 8 minutes for the Harvard images. For the AVIRIS data, this time was 12.53 minutes. The timing is for Matlab implementation on an Intel Core i7 CPU at 3.6 GHz with 8 GB RAM. For the Bayesian sparse coding stage, we again used $10^{-6}$ as the initial value for the parameters $a_o$ to $f_o$. We respectively initialized $\lambda_{s_o}$ and $\lambda_{e_o}$ to the final values of $\lambda_s$ and $\lambda_e$ of the dictionary learning stage. We ran the inferencing process $Q = 128$ times with 100 iterations in each run. It is worth noticing that, in the proposed sparse coding strategy, it is possible to run the inferencing processes independent of each other. This makes the sparse coding stage naturally suitable for multi-core processing. On average, a single sampling process required around 1.75 minutes for a CAVE image and approximately 7 minutes for a Harvard image. For the AVIRIS image, this time was 11.23 minutes.

The proposed approach outperforms the existing methods on ground based imagery as well as remotely sensed data, without requiring explicit parameter tuning. This distinctive characteristic of the proposed approach comes from exploiting the non-parametric Bayesian framework.

### 7.7 Conclusion

We proposed a Bayesian sparse representation based approach for hyperspectral image super resolution. Using the non-parametric Bayesian dictionary learning, the proposed approach learns distributions for the scene spectra and their proportions in the image. Later, this information is used to sparse code a high resolution image (e.g. RGB) of the same scene. For that purpose, we proposed a Bayesian sparse coding method that can be generically used with the dictionaries learned using the Beta process. Theoretical analysis is provided to show the effectiveness of the method. We used the learned sparse codes with the image spectra to construct the super resolution hyperspectral image. Exhaustive experiments on three public data sets show that the proposed approach outperforms the existing state of the art.
Hierarchical Beta Process with Gaussian Process prior for Hyperspectral Image Super-Resolution

Abstract

Hyperspectral cameras acquire precise spectral information, however, their resolution is very low due to hardware constraints. We propose an image fusion based hyperspectral super resolution approach that employs a Bayesian representation model. The proposed model accounts for spectral smoothness and spatial consistency of the representation by using Gaussian Processes and a spatial kernel in a hierarchical formulation of the Beta Process. The model is used by our approach to first infer Gaussian Processes for the spectra present in the hyperspectral image. Then, it is used to estimate the activity level of the inferred processes in a sparse representation of a high resolution image of the same scene. Finally, we use the model to compute multiple sparse codes of the high resolution image, that are merged with the samples of the Gaussian Processes for an accurate estimate of the high resolution hyperspectral image. We perform experiments with remotely sensed and ground-based hyperspectral images to establish the effectiveness of our approach.

Keywords: Hyperspectral, Super-resolution, Beta Process, Gaussian Process, Sparse representation.

8.1 Introduction

Spectral characteristics of materials are considered vital in remote sensing, medical imaging and forensics [69], [128], [241], [109], [24], [44]. Recently, they have also shown improved performance in various computer vision tasks, e.g. recognition [231], [201], [65], document analysis [112], [113], tracking [152], pedestrian detection [94] and segmentation [192]. Hyperspectral imaging is an emerging modality that can efficiently obtain high-fidelity spectral representations of a scene. Nevertheless, the low resolution of contemporary hyperspectral cameras is currently a bottleneck in its ubiquitous use [109], [47], [5].

Reflectance spectra are characterized by their intensity distributions over continuous wavelength ranges. Hence, hyperspectral cameras integrate scene radiance...
with hundreds of spectrally sharp bases, thereby requiring longer exposures. This results in a reduced resolution image. Moreover, it is not straightforward to use high resolution sensors in hyperspectral cameras because they further reduce the photon density that is already confined by the spectral filters. These constraints make hyperspectral image super resolution a particularly interesting research problem [8].

Currently, the resolution of the cameras that perform a gross quantization of the scene radiance (e.g. RGB and RGB-NIR), is orders of magnitude higher than that of hyperspectral cameras [109]. We collectively term the images acquired by these cameras as the multi-spectral images. In this work, we propose to take advantage of the high resolution of a multi-spectral image by merging its spatial patterns with the samples of Gaussian Processes [167], learned to represent the hyperspectral image of the same scene. Gaussian Processes provide an excellent tool for modeling natural spectra [66] because they can easily incorporate the regularly occurring smoothness of spectra. To learn the Gaussian Processes, we propose a novel Bayesian representation model. Our model also incorporates spatial consistency in the representation by employing a kernel in the hierarchical formulation of the Beta Process [193]. We provide a detailed Markov Chain Monte Carlo (MCMC) analysis [26] for the Bayesian inference using our model.

Employing the proposed model, we develop an approach for hyperspectral image super resolution, shown in Fig. 8.1. The approach first uses the model to infer Gaussian Processes to represent the low resolution hyperspectral image. Then, the mean parameters of the processes are transformed to match the spectral quantization of the high resolution multi-spectral image. The model is then employed with the transformed means and the multi-spectral image to infer a set of Bernoulli distributions. These distributions record the activity level of the Gaussian Processes in the representation of the multi-spectral image. Exploiting these distributions, the model is later applied to infer a set of sparse codes for the multi-spectral image, that are used with the samples of the Gaussian Processes to obtain a high resolution hyperspectral image. Experiments on remotely sensed and ground-based hyperspectral images show that our approach obtains higher fidelity super resolution images as compared to the state-of-the-art approaches.

8.2 Related Work

Hyperspectral imaging has been used in remote sensing for over three decades [185]. However, hyperspectral instruments installed in contemporary remote sensing platforms still lack in spatial resolution [24], [5]. High cost of replacing these instruments
Figure 8.1: Schematics: A set of Gaussian Processes (\( \mathcal{GP} \))s is inferred for the spectra in hyperspectral image. The means of the \( \mathcal{GP} \)s are transformed according to the spectral channels of a high resolution image \( \mathbf{X} \) of the same scene. The transformed means and \( \mathbf{X} \) are used to compute a set \( \mathcal{B} \) of Bernoulli distributions, signifying the activity level of \( \mathcal{GP} \)s in the sparse codes of \( \mathbf{X} \). Multiple sparse codes of \( \mathbf{X} \) are computed using the proposed model, each satisfying \( \mathcal{B} \). The computed codes are used with the samples of \( \mathcal{GP} \)s to estimate the high resolution hyperspectral image.

and hardware limitations have motivated significant research in signal processing based resolution enhancement of remote sensing imagery [5]. Pan-sharpening [10] is one of the common techniques used for this purpose. It fuses a pan-chromatic high resolution image with a hyperspectral image to improve its spatial resolution. Wavelet based pan-sharpening [154], Intensity-Hue-Saturation transform based methods [88], [37] and pan-sharpening with principal component analysis [180] are a few representative examples of this category.

Whereas sharp spatial patterns are apparent in pan-sharpened images, the resulting images often suffer from significant spectral distortions [39]. This can be attributed to the spectral limitations of the pan-chromatic images [118]. Therefore, Zhukov et al. [243] and Minghelli-Roman et al. [144] fused multi-spectral images with hyperspectral images. They used hyperspectral unmixing [110] for the image fusion. However, these methods assume relatively small difference between the spectral resolutions of the images being fused. Moreover, they also under-perform when the imaged scenes are highly mixed [91]. For such cases, their performance have been improved by Zurita-Milla et al. [244] by employing a sliding window technique.

Recently, matrix factorization based approaches have consistently shown state-
of-the-art performance in hyperspectral image super resolution. These approaches are divided into three categories based on their underlying assumptions. The methods in the first category \cite{109}, \cite{5}, \cite{8}, \cite{91} assume that only the spectral transform between the images being fused is known beforehand. Kawakami et al. \cite{109} factored a hyperspectral image and an RGB image into their respective bases and used the sparse codes of the RGB image with the basis of the hyperspectral image. Huang et al. \cite{91} applied a similar approach to remote sensing imagery, using singular value decomposition to learn the bases, whereas a sparsity controlled approach was employed in \cite{8} for non-negative matrix factorization. Motivated by the success of Bayesian matrix factorization in RGB and gray scale image super resolution \cite{89}, Akhtar et al. \cite{5} developed a Bayesian approach for hyperspectral image super resolution.

The approaches in the second category \cite{119}, \cite{120}, \cite{215}, \cite{210}, \cite{227} additionally assume priori knowledge of the spatial transform between the images. Lanaras et al. \cite{119}, \cite{120} formulated hyperspectral image super resolution as a coupled unmixing problem and proposed a matrix factorization algorithm to solve it. Their approach exploits the physical constraints generally followed by the spectral signatures. Wycoff et al. \cite{215} proposed to use the ADMM algorithm \cite{30} for the factorization procedure. A variational based approach is also proposed by Wei et al. \cite{210} for the same purpose. In \cite{227}, Yokoya et al. exploited the coupling between the images being fused and developed a coupled matrix factorization technique for hyperpsectral image super resolution. They also noted that in practice, it is challenging to obtain an accurate estimate of the spatial transform between the images being fused. Nevertheless, accurate transform is generally assumed and exploited by the approaches in this category for the improved performance. The methods in the third category \cite{118} assume availability of high resolution hyperspectral training data. This implicitly imposes both of the above assumptions in addition to the requirement that abundant training data is available despite hardware limitations.

Overall, the least restrictive methods are those belonging to the first category, in which our approach also falls. Akhtar et al. \cite{5} have recently established the usefulness of Bayesian sparse representation for hyperspectral image super resolution. Nevertheless, their approach has two major limitations. That is, it neither considers the spectral smoothness nor the spatial consistency of the representation of the nearby image pixels. In this paper, we address these limitations by 1) employing Gaussian Processes for the spectral signatures to incorporate smoothness into their representation; and 2) enforcing spatial consistency of the representation with a suitable kernel in the hierarchical formulation of the Beta Process.
8.3 Problem Formulation

Let us denote an acquired hyperspectral image with \( L \) spectral bands by \( X^h \in \mathbb{R}^{m \times n \times L} \). Let \( X \in \mathbb{R}^{M \times N \times l} \) be the high resolution image of the same scene obtained by a multi-spectral sensor. We aim at estimating a high resolution hyperspectral image \( H \in \mathbb{R}^{M \times N \times L} \) by merging \( X \) with \( X^h \). The two available images are considered to be linear mappings of the target image. Formally, \( X^h = \Omega^h(H) \) and \( X = \Omega(H) \), where \( \Omega^h : \mathbb{R}^{M \times N \times L} \rightarrow \mathbb{R}^{m \times n \times L} \) and \( \Omega : \mathbb{R}^{M \times N \times L} \rightarrow \mathbb{R}^{M \times N \times l} \). Moreover, we consider \( M \gg m, N \gg n \) and \( L \gg l \), to appropriately model the practical conditions. We neither assume prior knowledge of the spatial transform between the images being fused nor the availability of high resolution hyperspectral training data. Following [109], [5], [8], [91], and [120], we assume aligned \( X \) and \( X^h \). In practice, accurate alignment is possible using beam-splitting mechanism [94].

We denote the number of pure spectral signatures (a.k.a endmembers) in an imaged scene by \( K \) and the \( k \)th signature by \( \psi_k \in \mathbb{R}^L \). Let \( \Psi \in \mathbb{R}^{L \times K} \) be the matrix comprising these spectral signatures. Thus, a pixel \( x^h \in \mathbb{R}^L \) of the hyperspectral image can be represented as \( x^h = \Psi \alpha \), where \( \alpha \in \mathbb{R}^K \) is a coefficient vector. A pixel of the multi-spectral image can similarly be represented as \( x = \tilde{\Psi} \beta \), where \( \tilde{\Psi} \in \mathbb{R}^{l \times K} \) is obtained by transforming \( \Psi \), such that \( \tilde{\Psi} = \Delta \Psi \). Following the literature (see Section 8.2), we assume a priori knowledge of the spectral transformation operator \( \Delta \in \mathbb{R}^{l \times L} \). Since the exact value of \( K \) is generally unknown, we allow for \( K > L \). Hence, we expect the coefficient vectors \( \alpha \) and \( \beta \) to be sparse. Adopting the naming conventions from the sparse representation literature [176], we refer to \( \Psi \) and \( \tilde{\Psi} \) as dictionaries; to their columns as dictionary atoms; and the vectors \( \alpha \) and \( \beta \) as sparse codes.

8.4 Proposed Approach

Our approach utilizes a hierarchical Bayesian sparse representation model, that we propose to represent hyperspectral images. The model is used to infer an ensemble of dictionaries for the hyperspectral image in the form of Gaussian Processes (\( \mathcal{GP} \))s [167]. The approach transforms the mean parameters of the \( \mathcal{GP} \)s and uses them with the multi-spectral image \( X \) to estimate the activity level of the \( \mathcal{GP} \)s in the sparse representation of \( X \). We again utilize the proposed model for this estimation. Lastly, the model is used to learn multiple sparse codes of \( X \) that are combined with

\footnote{In Appendix C.3, we also demonstrate the robustness of our approach to image misalignment up to 10 pixels.}
the samples from the $\mathcal{GP}$s to compute the high resolution hyperspectral image. The proposed approach is summarized in Fig. 8.1.

8.4.1 Dictionary Learning

Below, we describe the proposed representation model and its Bayesian inference process that results in the learning of the $\mathcal{GP}$s (i.e. dictionary atoms). Other stages of our approach also exploit the same model, with minor variations in the inference process. We explain those variations in Sections 8.4.2 and 8.4.3.

Representation model: We model the $i$th pixel of a hyperspectral image as

$$x_i^h = \Psi \alpha_i + \epsilon_i^h,$$

where $\epsilon_i^h \in \mathbb{R}^L$ represents noise. The following hierarchical Bayesian representation model is proposed to compute the probability distributions over the dictionary atoms and the sparse codes:

$$x_i^h = \Psi \alpha_i + \epsilon_i^h \tag{8.1}$$

In the above expressions, $\mathcal{N}$, $\text{Gam}$, $\text{Bern}$ and $\text{Beta}$ respectively denote the Normal, Gamma, Bernoulli and the Beta probability distributions. The symbol $\odot$ denotes the element-wise product and the subscript ‘$o$’ signifies the hyper-parameters of the distributions whose values remain fixed during the inference process (discussed below). For reading convenience, we explain the remaining symbols along the relevant discussion on the representation model.

In the proposed model, we let the $k$th dictionary atom $\psi_k$ to be a sample drawn from a Gaussian Process. We define the kernel of a Gaussian Process, i.e. $\Sigma_k \in \mathbb{R}^{L \times L}$, such that it promotes high correlations between the adjacent coefficients of $\psi_k$. Recall that, $\psi_k$ signifies a spectral signature in our formulation. Thus, the kernel incorporates the relative smoothness of the spectra in the proposed model. In the given definition of the kernel, $\theta_t$ denotes the wavelength at the $t$th channel of the image, whereas $|$ represents the absolute value. Such an exponential form of the kernel is common for Gaussian Processes [167]. However, we also include an additional
scaling parameter $\eta_k$ in the kernel to allow it to adjust to the observed data. The value of this parameter is automatically inferred in our approach. We place a non-informative Gamma prior over $\eta_k$; such that $\text{Gam}(\eta_k|a_0,b_0) = \frac{b_0^{a_0} e^{-b_0 \eta_k}}{\Gamma(a_0)}$, where $\Gamma(.)$ is the well-known Gamma function. The remaining model also utilizes the same functional form of the Gamma prior. In our approach, the value of $\eta_0$ is fixed to $1/L$.

We compute the $k$th coefficient $\alpha_{ik}$ of $\alpha_i$ as the product of a sample $w_{ik}$ from a Normal distribution, with precision $\lambda_w$; and a sample $z_{ik} \in \{0,1\}$ from a Bernoulli distribution, with parameter $0 \leq \pi_{ik} \leq 1$. Thus, according to our model, a pixel $x^h_i$ selects the $k$th dictionary atom in its representation with a probability $\pi_{ik}$. For a similar situation, Akhtar et al. [5] directly placed a Beta probability prior over the Bernoulli distribution parameter. However, their approach neither forces the atoms of the dictionary nor the sparse codes to be similar for the nearby pixels in the image. To enforce this spatial consistency in the representation model, we compute $\pi_{ik}$ as a weighted sum of the samples from a Beta probability distribution. In our approach, these Beta distribution samples signify the probabilities of selection of $\psi_k$ in the representations of the nearby pixels of $x^h_i$. Hence, $\psi_k$ has more chances to get selected for $x^h_i$, if that dictionary atom is also used in the representations of the nearby pixels of $x^h_i$.

Concretely, we let $\pi_{ik} = e^T (\kappa \circ \Xi_{ik}) e$, where $e \in \mathbb{R}^P$ is a vector of 1s; $\kappa \in \mathbb{R}^{P \times P}$ is the spatial kernel and $\Xi_{ik} \in \mathbb{R}^{P \times P}$ comprises the samples of a Beta probability distribution. Here, $P$ is the size of the image patch that contains the neighborhood pixels centered around $x^h_i$. We compute a coefficient of $\kappa$ at index $(q,r)$ as $\kappa(q,r) = \exp(-||I_t - I_j||_2/\sigma_o)$, where $I_t$ denotes the index of the $t$th pixel in the image and $\sigma_o$ decides the kernel width. We sample $\Xi_{ik}(q,r)$ from a Beta distribution and, keeping in view the physical significance of these samples, we place a second Beta prior over the parameter $\rho_k$ of the distribution. The second prior plays the same role in our model that is played by the Beta prior in the model employed by Akhtar et al. [5]. However, the resulting stochastic process uses a Gaussian Process as the base measure in our model, instead of a Multi-variate Gaussian, as in [5]. We also note that the notion of hierarchical construction of the Beta Process was first introduced by Thibaux and Jordan [193]. However, our model differs from their proposal, as they did not use a kernel for computing $\pi_{ik}$ and employed a Normal distribution as the base measure.

Following the literature [109], [5] we consider white noise in our model\textsuperscript{2}. In

\textsuperscript{2}The model can be readily extended to handle colored noise. For that, instead of isotropic
Eq. 8.1, the covariance matrix of the noise distribution is denoted as \( \lambda_c^{-1} I_L \), where \( I_L \in \mathbb{R}^{L \times L} \) is the identity matrix. We place a Gamma prior over the noise precision \( \lambda_c \). This allows a Bayesian model to automatically adjust to the noise level of the observed data [237].

Inference: We perform Markov Chain Monte Carlo (MCMC) analysis [26] to infer the posterior probability distributions over the dictionary atoms and the sparse codes using our model. Below, we derive the expressions of the probability distributions that are sampled sequentially to perform the MCMC analysis. The sampling process is carried out iteratively.

**Sampling \( \psi_k \):** For brevity, let us denote the contribution of \( \psi_k \) to \( x_i^h \) as \( x_{i\psi_k}^h = x_i^h - \Psi(w_i \odot z_i) + \psi_k(w_ikz_ik) \), where \( w_i, z_i \in \mathbb{R}^K \) are the vectors formed by concatenating \( w_ik \) and \( z_ik \), \( \forall k \). According to our model, the posterior probability distribution over \( \psi_k \) can be expressed as:

\[
p(\psi_k | \cdot)=\prod_{i=1}^{mn} \mathcal{N}(x_{i\psi_k}^h | \psi_k(w_ikz_ik), \lambda_c^{-1} I_L)\mathcal{GP}(\psi_k|0, \Sigma_k).
\]

Exploiting the linear Gaussian model [175], it can be shown that \( \mathcal{GP}(\psi_k|\mu_k, \hat{\Sigma}_k) \) must be used to sample this posterior probability distribution, where

\[
\hat{\Sigma}_k = \left( \Sigma_k^{-1} + \lambda_c \sum_{i=1}^{mn} (w_ikz_ik)^2 \right)^{-1}, \quad \mu_k = \lambda_c \hat{\Sigma}_k \sum_{i=1}^{mn} (w_ikz_ik)x_{i\psi_k}^h.
\]

**Sampling \( \eta_k \):** According to the proposed model, the posterior probability distribution over \( \eta_k \) can be written as \( p(\eta_k|\cdot) \propto \mathcal{GP}(\Psi|0, \frac{1}{\eta_k} \hat{\Sigma}_k)\operatorname{Gam}(\eta_k|a_o, b_o) \), where \( \Psi \) is a vector of zeros in \( \mathbb{R}^L \) and \( \hat{\Sigma}_k = \exp(-|\theta_b - \theta_a|/\eta_o) \). The right hand side of the proportionality can be further be expanded into the following expression:

\[
\frac{1}{\sqrt{2\pi}^{L/2}} \eta_{\text{o}}^{L-1} \Gamma(a_o)^{\frac{1}{2}} \det(\frac{1}{\eta_k} \hat{\Sigma}_k)^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} \eta_k^{-1} \psi_k^T \hat{\Sigma}_k \psi_k - b_o \eta_k \right\},
\]

which is proportional to the Gamma probability distribution \( \operatorname{Gam}(\eta_k|a_o + \frac{L}{2}, b_o + \frac{1}{2} \psi_k^T \hat{\Sigma}_k \psi_k) \), that we use to sample \( \eta_k \).

**Sampling \( w_ik \):** The posterior distribution over \( w_ik \) has the functional form \( p(w_ik|\cdot) \propto \mathcal{N}(x_{i\psi_k}^h | \psi_k(w_ikz_ik), \lambda_c^{-1} I_L) \mathcal{N}(w_ik|0, \lambda_w^{-1}) \). Again, making the use of the linear Gaussian model, \( w_ik \) can be sampled from \( \mathcal{N}(w_ik|\mu_w, \hat{\lambda}_w^{-1}) \), where \( \hat{\lambda}_w = \lambda_w + \lambda_c z_{ik}^2 \psi_k^T \psi_k \) and \( \mu_w = z_{ik} \psi_k^T x_{i\psi_k}^h \lambda_c/\lambda_w \).

**Sampling \( \lambda_w \):** According to our model, the posterior over \( \lambda_w \) can be written as:

\[
\frac{1}{\sqrt{2\pi}^{L/2}} \eta_{\text{o}}^{L-1} \Gamma(a_o)^{\frac{1}{2}} \det(\frac{1}{\eta_k} \hat{\Sigma}_k)^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} \eta_k^{-1} \psi_k^T \hat{\Sigma}_k \psi_k - b_o \eta_k \right\},
\]

covariance of the noise distribution, we can use a diagonal noise covariance matrix, with its \( i^{th} \) diagonal entry denoting the noise variance at the \( i^{th} \) band.
p(λ_w|−) ∝ \prod_{i=1}^{mn} N(\mathbf{w}_i|0, \lambda^{-1}_w I_K) \text{Gam}(\lambda_w|c_o, d_o). \quad \text{Hence, employing the conjugacy between the Normal and the Gamma distributions, we sample } \lambda_w \text{ from } \text{Gam}(\lambda_w|\frac{Kmn}{2} + c_o, \frac{1}{2} \sum_{i=1}^{mn} \mathbf{w}_i^T \mathbf{w}_i + d_o). \\

\textbf{Sampling } z_{ik}: \text{ We can write } p(z_{ik}|−) \propto N(x_{w_{ik}}^h|\psi_k(w_{ik}z_{ik}), \lambda^{-1}_w I_L) \text{Bern}(z_{ik}|\tau_{ik}). \quad \text{From here, it is easy to show that } z_{ik} \text{ must be sampled from the Bernoulli distribution: } \text{Bern}\left(z_{ik}\left|\frac{\gamma}{1-\gamma}\right\right), \text{ where }\gamma = \exp\left(-\frac{1}{2}\left(\psi_k^T \psi_k w_{ik}^2 - 2w_{ik} x_{w_{ik}}^T \psi_k\right)\right). \\\n
\textbf{Sampling } \Xi_{ik}(q,r): \text{ Let } \Xi_{ik}(q,r) = \xi_{ik} \text{ and } \tilde{\pi}_{ik} \sim \pi_{ik} - \kappa(q,r) \xi_{ik}. \text{ Using these notations, we can express the posterior distribution over } \Xi_{ik}(q,r) \text{ as:} \\
p(\xi_{ik}|−) \propto \text{Beta}(\xi_{ik}|c_o \rho_k, f_o(1 - \rho_k)) \prod_{q,r \in \{1, ..., P\}} \text{Bern}\left(z_{ik}|\kappa(q,r) \xi_{ik} + \tilde{\pi}_{ik}\right). \\

This distribution can not be directly sampled. However, considering its functional form, it is possible to associate the popularity of the } k^{th} \text{ dictionary atom to the pixels in the neighborhood } \mathcal{K} \text{ of } \mathbf{x}_i^h \text{ as:} \\
\xi_{ik} \sim \text{Beta}\left(\xi_{ik}|\xi_{i} + \sum_{(j:p,q) \in \mathcal{K}} z_{jk}, f_o(1 - \rho_k) + \sum_{(j:p,q) \in \mathcal{K}} (1 - z_{jk})\right). \\

We use the above distribution as the proposal distribution } Q \text{ in the Metropolis Hastings (MH) algorithm [86] and in step } \tau \text{ of MH algorithm, we draw } \xi_{ik} \sim Q(\xi_{ik}|\xi_{ik}) \text{, where } \xi_{ik} \text{ is the current } \xi_{ik}, \text{ and accept the sample with a probability:} \\
\vartheta = \min\left\{1, \frac{p(\xi_{ik}) Q(\xi_{ik}|\xi_{ik}^*)}{p(\xi_{ik}^*) Q(\xi_{ik}|\xi_{ik})}\right\} \\

It can be shown that the fraction in the brackets can be analytically computed as follows: \\
\left(\frac{\xi_{ik}^*}{\xi_{ik}}\right) \sum_{(j:p,q) \in \mathcal{K}} \frac{z_{jk}}{1 - \xi_{ik}^*} \left(1 - \frac{\xi_{ik}^*}{\xi_{ik}}\right) \sum_{(j:p,q) \in \mathcal{K}} \frac{(1 - z_{jk})}{1 - \xi_{ik}^*} \prod_{(j:p,q) \in \mathcal{K}} \left(1 + \frac{\Upsilon}{\xi_{ik}^*}\right) \frac{z_{jk}}{\xi_{ik}^*} \left(1 - \frac{\Upsilon}{1 - \xi_{ik}^*}\right)^{1 - z_{jk}}, \\

where } \Upsilon = \kappa(p,q)(\xi_{ik}^* - \xi_{ik}). \\

\textbf{Sampling } \rho_k: \text{ The posterior on } \rho_k \text{ can be expressed as:} \\
p(\rho_k|−) \propto \text{Beta}(\rho_k|\gamma_o/K, \gamma_o(K - 1)/K) \prod_{i=1}^{mn} \text{Beta}(\xi_{ik}|\xi_{i} + \sum_{(j:p,q) \in \mathcal{K}} z_{jk}, f_o(1 - \rho_k)). \\

For analytical simplification, we let } e_o = f_o = 1. \text{ By expanding the expressions for the distributions and neglecting the constant terms, we can show that } \forall i \in \{1, ..., mn\}: \\
p(\rho_k|−) \propto \rho_k^{\frac{\gamma_o}{K} - 1} (1 - \rho_k)^{\frac{\gamma_o(K - 1)}{K} - 1} \left(\frac{\xi_{ik}}{1 - \xi_{ik}}\right)^{p_k mn} \rho_k^{mn}. 

Here, the term depending on $\xi_{ik}$ can be simplified to $\exp\left(\rho_k \sum_{i=1}^{mn} \log \frac{\xi_{ik}}{1 - \xi_{ik}}\right)$. Therefore, following Zhou et al. [239], we use slice sampling algorithm [62] to sample $\rho_k$ from the following exponential distribution:

$$\rho_k \sim \text{Exp}\left(\sum_{i=1}^{mn} \log \frac{\xi_{ik}}{1 - \xi_{ik}}\right) R(\varsigma, \upsilon, \omega),$$

where $R(\varsigma, \upsilon, \omega)$ is the range of $\rho_k$ and $\upsilon \sim \text{Unif}(0, (1 - \rho_k)(\frac{\sin^{mn}(\pi \rho_k)}{\rho_k})^{-1})$, $\varsigma \sim \text{Unif}(0, \rho_k)$, and $\omega \sim \text{Unif}(0,\sin^{mn}(\pi \rho_k))$. Here, Unif denotes the uniform distribution. We restrict $0 < \rho_k < 1$ and exploit the fact that $\Gamma(\rho_k)\Gamma(1 - \rho_k) \propto 1/\sin(\pi \rho_k)$ [87] to arrive at the expression for sampling $\omega$.

**Sampling $\lambda$:** We have $p(\lambda_i|-) \propto \prod_{i=1}^{mn} N(x_i^h|\Psi(w_i \odot z_i), \lambda_i^{-1}I_L)\text{Gam}(\lambda_i|k_o, l_o)$. Again, by employing the conjugacy of the probability distributions we can sample $\lambda_i$ from $\text{Gam} \left(\lambda_i^{-1}Lmn + k_o, \frac{1}{2} \sum_{i=1}^{mn} \|x_i^h - \Psi(w_i \odot z_i)\|_2^2 + l_o \right)$, where $\|.|_2$ denotes the vector $l$-2 norm.

### 8.4.2 Support Distribution Learning

Once the inference is complete, we get a set $G \subset \mathbb{R}^L$ of $K$ Gaussian Processes, where each process represents a probability distribution over a dictionary atom. Probability distributions over other model parameters (e.g. $z_{ik}$) are also inferred as by-products, however they are not required by our approach. We transform the means of the $\mathcal{GP}$s using the spectral transformation operator $\Delta$ and use them to represent the high resolution multi-spectral image $X$. To learn the representation, we again use the proposed model. However, during inference, instead of sampling for the dictionary atoms, we keep them fixed to the transformed means of the $\mathcal{GP}$s. For our model, this implies $\eta_k \rightarrow \infty$. Therefore, we also do not sample for $\eta_k$. These modifications in the sampling process effectively reduce our dictionary learning process to a sparse coding process. We defer further discussion on sparse coding to Section 8.4.3. Here, we are interested in the set $\mathcal{B} \subset \mathbb{R}$ of $K \times MN$ Bernoulli distributions (i.e. parameters $\pi_{ik}, \forall i, k$) computed by the inference process. This set contains $K$ distributions for each pixel of $X$ that determines the support (indices of non-zero elements) of the sparse codes for that pixel. Since the basis vectors for the sparse codes are the transformed means of the $\mathcal{GP}$s, $\mathcal{B}$ encodes the activity level of the $\mathcal{GP}$s in the sparse representation of $X$. We store $\mathcal{B}$ to later exploit it in an accurate reconstruction of the high resolution hyperspectral image.

We emphasize that the association of $K$ Bernoulli distributions with a single pixel in our approach is different from the affiliation of $K$ such distributions with the com-
8.4.3 Sparse Coding

Let us briefly consider the sparse coding process for a pixel $x$ of $X$, discussed in the previous section. That process computes the codes $\beta$ of $x$, such that their support follows the Bernoulli distributions in $B$. Let $z \in \mathbb{R}^K$ be the binary vector indexing that support. It is easy to imagine that sampling the same distributions in $B$ multiple times, can often result in different $z$. Being a sample of the learned Bernoulli distributions, each such $z$ is a useful support of the sparse codes for our approach. This entails the existence of multiple useful sparse codes and hence, multiple useful reconstructions of $x$ in our probabilistic settings.

Let $\tilde{\Psi}$ denote the dictionary formed by the transformed means of the GPs. We can write, $\tilde{x} = \tilde{\Psi}(w \odot z)$, where $\beta = w \odot z$ and $\tilde{x}$ is the reconstructed $x$. We propose the following lemma regarding $\tilde{x}$:

**Lemma 3.** $\|\mathbb{E}[\tilde{x}] - x\|^2_2 \leq \|\tilde{x} - x\|^2_2$, where $\mathbb{E}[-]$ is the expectation operator.

**Proof:** We can write $\|\mathbb{E}[\tilde{x}] - x\|^2_2 = \|\tilde{\Psi}\mathbb{E}[\beta] - x\|^2_2$. Since $\beta = w \odot z$ and multiple $z$ exist, we can exploit the conditional expectation of the discrete random variables to write $\mathbb{E}[\beta] = \mathbb{E}[\mathbb{E}[\beta|z]]$ [174]. From the results in [5], we already know that $\mathbb{E}[\mathbb{E}[\beta|z]] = \beta_{opt}$, where $\beta_{opt}$ is the optimal $\beta$ with respect to the squared error. Since, $\mathbb{E}[\tilde{x}] = \tilde{\Psi}\beta_{opt}$ and $\tilde{x} = \tilde{\Psi}\beta$, where $\beta$ is not guaranteed to be optimal, $\|\mathbb{E}[\tilde{x}] - x\|^2_2 \leq \|\tilde{x} - x\|^2_2$.

Lemma 3 shows that the expected value of multiple reconstructions of $x$ can be superior to its single reconstruction. Moreover, by using $\tilde{\Psi} = \Delta \tilde{\Psi}$ in the above proof, we can extend this result to show that $\|\mathbb{E}[\tilde{h}] - h\|^2_2 \leq \|\tilde{h} - h\|^2_2$, where $h$ and $\tilde{h}$ denote the pixels of the target high resolution hyperspectral image and its reconstruction, respectively. To exploit this finding, we adopted the following strategy for the reconstruction of the super resolution image $H$. First, we compute $Q$ sparse codes for $X$ using our model. For these computations, we fixed both $\tilde{\Psi}$ and $B$ during the Bayesian inference. Since the sparse codes in Section 8.4.2 were also computed using the same $B$ and $\tilde{\Psi}$, we also use them in the upcoming computations. Second, we draw $Q + 1$ samples from the already inferred Gaussian Processes and use them with the available $Q + 1$ sparse codes to construct the same number of reconstructions of $H$. Finally, we estimate the expected value of these $Q + 1$ reconstructions by computing their mean.
8.5 Experiments

We have evaluated our approach on both remote sensing and ground-based hyperspectral images. For the evaluation metrics, we used the Root Mean Squared Error (RMSE) [5] and the Spectral Angle Mapper (Sam) [229]. We mainly compare our approach to the Matrix Factorization based approach (MF) [109], the Spatial-Spectral Fusion Method (SSFM) [91], the Generalized Simultaneous OMP based method (GSOMP) [8] and the Bayesian Sparse Representation approach (BSR) [5]. These are the state-of-the-art approaches in the first category of the hyperspectral super-resolution techniques (see Section 8.2) to which our approach also belongs. We note that few approaches from the second and the third category have recently reported impressive results [227], [120], [118]. However, those results are obtained by exploiting additional prior knowledge, which is not always available (and hence, not assumed in this work).

In our experiments, we used the author-provided implementations for GSOMP and BSR, with the parameter values reported for the same data sets in [8] and [5]. Due to the unavailability of public codes for MF and SSFM, we implemented these approaches using the SPAMS library [136], that is well-know for its accuracy. The parameter values of these approaches were carefully optimized such that the achieved results were the same or better than the previously reported best results for these approaches on common images. We defer the discussion on the parameter settings of our approach to Section 8.6. We follow a common evaluation protocol [8], [5], [109] that considers an available hyperspectral image as the ground truth and constructs a low resolution hyperspectral image by averaging $32 \times 32$ disjoint blocks of the ground truth. A high resolution multi-spectral image is constructed by spectral transformation of the ground truth, with a known transformation operator $\Delta$.

For the remote sensing images, we used a data set provided by NASA\(^3\), that contains four hyperspectral images collected by the airborne sensor AVIRIS [80]. These $512 \times 512 \times 224$ images are acquired in the wavelength range 370-2500 nm, over Cuprite mines in Nevada, US. Due to water absorptions and low signal-to-noise ratio, we removed 36 channels from the images, corresponding to the wavelengths 370, 380, 1330 to 1430, 1780 to 1970, 2490 and 2500 nm. The resulting images are considered as the ground truth. We constructed a high resolution multi-spectral image by selecting six bands of the ground truth, corresponding to the wavelengths 480, 560, 660, 830, 1650 and 2220 nm. These bands roughly correspond to the visible and mid-infrared wavelength channels of NASA-Landsat 7 satellite. Thus,

\(^3\)Download link: ftp://popo.jpl.nasa.gov/pub/free_data/f970619t01p02r02c_rfl.tar.
Table 8.1: Benchmarking on remote sensing images: The results are in the range of 8 bit images. The best results are given in bold in green cells. The second best values are in blue cells. Image names are according to the source data set.

<table>
<thead>
<tr>
<th>Method</th>
<th>SC01 RMSE</th>
<th>SC01 Sam</th>
<th>SC02 RMSE</th>
<th>SC02 Sam</th>
<th>SC03 RMSE</th>
<th>SC03 Sam</th>
<th>SC04 RMSE</th>
<th>SC04 Sam</th>
</tr>
</thead>
<tbody>
<tr>
<td>MF [109]</td>
<td>1.32</td>
<td>1.85</td>
<td>1.55</td>
<td>1.60</td>
<td>1.62</td>
<td>1.51</td>
<td>2.73</td>
<td>2.49</td>
</tr>
<tr>
<td>SSFM [91]</td>
<td>1.35</td>
<td>1.68</td>
<td>1.56</td>
<td>1.59</td>
<td>1.77</td>
<td>1.59</td>
<td>2.68</td>
<td>2.31</td>
</tr>
<tr>
<td>GSOMP [8]</td>
<td>1.30</td>
<td>1.39</td>
<td>1.52</td>
<td>1.63</td>
<td>1.79</td>
<td>1.80</td>
<td>1.54</td>
<td>2.05</td>
</tr>
<tr>
<td>BSR [5]</td>
<td>1.21</td>
<td>1.33</td>
<td>1.54</td>
<td>1.61</td>
<td>1.46</td>
<td>1.58</td>
<td>1.62</td>
<td>1.77</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.92</td>
<td>1.26</td>
<td>1.32</td>
<td>1.47</td>
<td>1.11</td>
<td>1.35</td>
<td>1.36</td>
<td>1.54</td>
</tr>
</tbody>
</table>

Figure 8.2: Effect of spectral smoothness and spatial consistency: Two contiguous ground truth pixels for SC01 are shown along their estimates with BSR [5] and the proposed approach.

the spectral transform $\Delta \in \mathbb{R}^{6 \times 188}$ is a binary matrix in this experiment, similar to [8], [5] and [91].

Comparison of performance on the remote sensing images is summarized in Table 8.1. The results are computed using the 8-bit intensity range. As visible in the table, a considerable improvement in the results is achieved by our approach. On average, the RMSE values for our approach are $\sim 18\%$ better than the previous lowest values. Similarly, the average gain in the spectral angle mapper values is $\sim 9\%$. This gain can be attributed to the spectral smoothness and the spatial consistency of the images reconstructed by our approach, which are common attributes of remote sensing hyperspectral images. In Fig. 8.2, we compare the reconstructions of two randomly selected contiguous pixels of image SC01 by BSR [5] and our approach. BSR showed the second best results on this image. It is easy to see that our approach
Figure 8.3: Spectral images for SC01 at 460, 540, 620 and 1500 nm: The $512 \times 512$ reconstructions of the ground truth are shown along the used $16 \times 16$ low resolution images. Absolute differences between the reconstructions and the ground truth are also shown. The absolute difference images for BSR [5] are included for comparison.

not only reconstructs each pixel better due to the spectral smoothness, but also due to the similarities between the adjacent pixels. The angles (in $\mathbb{R}^{188}$) between the shown ground truth pixels is $1.26^\circ$. For our approach, this angle is $1.28^\circ$, whereas for BSR, its value is $3.23^\circ$. We also show examples of the reconstructed spectral images for SC01 by our approach and BSR in Fig. 8.3. Figures for the other images are provided in Appendix C.2.

For the ground-based images, we evaluated our approach on hyperspectral images of everyday objects from the CAVE database [226] and the images of indoor and outdoor scenes from the Harvard database [40]. The $512 \times 512 \times 31$ images of the CAVE database are acquired using tunable filters over the wavelength range 400-700 nm. The fixed focal length of the sensor has resulted in a blur for the first two spectral bands of the images (illustrated in Appendix C.4). We removed these bands in our experiments to avoid any bias in the results. The remaining images are considered as the ground truth. The use of tunable filters in the Harvard database
Table 8.2: Benchmarking on ground-based images: The results are in the range of 8 bit images. CMF [227] and TDGU [118] (in red) are included for reference only, as these approaches belong to different categories and are not directly comparable to the other methods. For the rest, the best results are given in bold in green cells. The second best values are in blue cells.

<table>
<thead>
<tr>
<th>Image</th>
<th>CAVE database [226]</th>
<th>Harvard database [40]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Balloons</td>
<td>Beads</td>
</tr>
<tr>
<td><strong>Method</strong></td>
<td>RMSE</td>
<td>Sam</td>
</tr>
<tr>
<td>MF [109] (Cat. 1)</td>
<td>2.3</td>
<td>8.0</td>
</tr>
<tr>
<td>SSFM [91] (Cat. 1)</td>
<td>2.4</td>
<td>8.4</td>
</tr>
<tr>
<td>GSOMP [8] (Cat. 1)</td>
<td>2.3</td>
<td>8.1</td>
</tr>
<tr>
<td>BSR [5] (Cat. 1)</td>
<td>2.1</td>
<td>7.9</td>
</tr>
<tr>
<td>Proposed (Cat. 1)</td>
<td><strong>1.9</strong></td>
<td><strong>7.6</strong></td>
</tr>
<tr>
<td>CMF [227] (Cat. 2)</td>
<td>2.9</td>
<td>4.3</td>
</tr>
<tr>
<td>TDGU [118] (Cat. 3)</td>
<td>1.6</td>
<td>-</td>
</tr>
<tr>
<td>MF [109] (Cat. 1)</td>
<td>2.6</td>
<td>2.7</td>
</tr>
<tr>
<td>SSFM [91] (Cat. 1)</td>
<td>3.1</td>
<td>2.8</td>
</tr>
<tr>
<td>GSOMP [8] (Cat. 1)</td>
<td>3.3</td>
<td>2.9</td>
</tr>
<tr>
<td>BSR [5] (Cat. 1)</td>
<td>2.4</td>
<td>2.9</td>
</tr>
<tr>
<td>Proposed (Cat. 1)</td>
<td><strong>2.2</strong></td>
<td><strong>2.5</strong></td>
</tr>
<tr>
<td>CMF [227] (Cat. 2)</td>
<td>2.3</td>
<td>2.4</td>
</tr>
<tr>
<td>TDGU [118] (Cat. 3)</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
has resulted in spatial distortions in some images with moving objects (e.g. grass, trees). We also avoid these images in our experiments for a fair evaluation. Following [5], [8] we used the top-left $1024 \times 1024$ spatial patches as the ground truth.

Following [109], [5], [8] we constructed the high resolution multi-spectral images by transforming the ground truth with the spectral response of Nikon D700 camera (http://www.maxmax.com/spectral_response.htm). In Table 8.2, we show the results on five commonly used benchmarking images of each database. For reference, we also include results of representative methods from the remaining two categories of the matrix factorization based hyperspectral super-resolution techniques. The results of Coupled Matrix Factorization (CMF) [227] based approach are directly taken from [120], whereas the performance of the Training Data Guided Up-sampling (TDGU) method are taken from [118]. In Fig. 8.4, we show examples of the reconstructed super resolution spectral images by our approach. Further results and comparisons are also provided in Appendix C.1. Although the performance of the proposed approach is generally better than the existing approaches in the same category, the improvements in the results are not as significant as for the remote sensing images. In our opinion, the lower spectral resolution and larger variations in the spatial patterns in the ground-based images are the reasons behind this phenomenon. Nevertheless, our approach is generally able to perform better than the existing approaches on the ground-based images as well.

### 8.6 Discussion on Parameters

In all the experiments we used the value $10^{-6}$ for $a_o, b_o, c_o, d_o, k_o, l_o$ and 1 for $e_o$ and $f_o$. The values of $g_o$ and $h_o$ were adjusted to give a parabolic probability density function of the Beta distributions, for which $g_o/K \approx h_o(K - 1)/K$. We used $\eta_o = 1/L$ and the spatial kernel width was set to 2 for the Harvard database and 1 for the remaining data sets. This resulted in $P = 5$ and $P = 3$ respectively. Except for $P$, our model is fairly insensitive to small perturbations in the parameter values, which is a common observation for Bayesian models [5]. The reported results are sensitive to the value of $P$ because the considered low resolution hyperspectral images have very small spatial dimensions. Due to the spatial dimensions of the images, we used $K = 100$ for Harvard database and 10 for the remaining data sets.

We initialized $\lambda_c$ to $10^6$, $\lambda_w$ to $10^3$, $\eta_k$ to $10^{-3}, \forall k$ and $\pi_{ik}$ to $10^{-3}, \forall i, k$. These initial values were selected considering the physical significance of the parameters in our model. Nevertheless, the approach is generally insensitive to these initial values. To learn the Gaussian Processes, we initialized the dictionary with random
Figure 8.4: Spectral reconstruction at 460, 540 and 620nm: The used 16×16 low resolution images are shown along the reconstructions and the 512×512 ground truth. Absolute differences between the reconstructions and the ground truth are also given. (Left) ‘Balloons’ from the CAVE database [226]. (Right) ‘Img h0’ form the Harvard database [40].

samples of Multi-variate Gaussians and initialized the spare codes by allowing half of them to have value 1. For sparse coding, we used the LASSO solver of the SPAMS library [136] to initialize the sparse codes. In our experiments, we processed the images as 2×2 overlapping patches.

We used 500 sampling iterations for dictionary learning and 300 and 100 iterations respectively for learning the Bernoulli distributions in $B$ and sparse coding. Fewer iterations were enough in the later stages because fewer probability distributions were required to be sampled in those stages. We computed the codes 25 time, i.e. $Q = 25$ in our experiments.

8.7 Conclusion

We proposed a Bayesian approach for hyperspectral image super resolution that fuses a high resolution multi-spectral image with a hyperspectral image. Our approach utilizes a Bayesian sparse representation model that places Gaussian Process priors on the dictionary atoms and uses a kernel to promote spatial consistency in the representation. We proposed this model and derived its inference equations. The model is used in our approach for inferring Gaussian Processes for the dictionary atoms, estimating their popularity in the sparse representation of the multi-spectral
image and computing multiple sparse codes of that image. The sparse codes of the multi-spectral image are used with the samples of the Gaussian Processes to finally estimate the super resolution hyperspectral image. We tested our approach using remote sensing and ground-based hyperspectral images. Our results show that the approach is useful for both types of images, especially for the remote sensing images that generally comprise smoother spatio-spectral patterns.
PART III

Classification
CHAPTER 9

Efficient Classification with Sparsity Augmented Collaborative Representation

Abstract

Many classification approaches first represent a test sample using the training samples of all the classes. This collaborative representation is then used to label the test sample. It is a common belief that sparseness of the representation is the key to success for this classification scheme. However, more recently, it has been claimed that it is the collaboration and not the sparseness that makes the scheme effective. This claim is attractive as it allows to relinquish the computationally expensive sparsity constraint over the representation. In this paper, we first extend the analysis supporting this claim and then show that sparseness explicitly contributes to improved classification, hence it should not be completely ignored for computational gains. Inspired by this result, we augment a dense collaborative representation with a sparse representation and propose an efficient classification method that capitalizes on the resulting representation. The augmented representation and the classification method work together meticulously to achieve higher accuracy and lower computational time compared to state-of-the-art collaborative representation based classification approaches. Experiments on benchmark face, object and action databases show the efficacy of our approach.

Keywords: Multi-class classification, Sparse representation, Collaborative representation.

9.1 Introduction

Several recent approaches for multi-class classification (e.g. [63], [104], [106], [165], [188], [213], [221], [224], [233]) exploit the representation of a test sample over a redundant basis, formed by the training samples (or their extracted features). This collaborative representation of the test sample, in which the training samples from different classes collaborate to approximate the test sample, is later used to decide its class label. Wright et al. [213] first demonstrated the impressive potential of this scheme for face recognition. Their approach additionally forces the representation

\(^9\)Under review by Pattern Recognition (PR) journal.
to be sparse (i.e. it uses only a few vectors from the basis). Hence, it is called Sparse Representation based Classification (SRC).

The success of SRC was followed up by its variants. For instance, Huang et al. [92] proposed a transformation-invariant SRC. Zhou et al. [242] combined Markov Random Fields with SRC for disguised faces. Similarly, Wagner et al. [202] enhanced SRC for the misalignment, pose and illumination invariant recognition. Yang et al. [223] proposed a robust sparse representation technique to be used for face recognition. Effectiveness of these approaches also boosted significant research in dictionary learning [71] based multi-class classification [7], [104], [137], [163], [166], [191]. Initially, the success of these approaches was attributed to the sparseness of the used representation. However, more recently, researchers have started questioning the role of sparsity in such approaches [170], [182], [233]. Among them, Zhang et al. [233] analyzed the working mechanism of SRC and claimed that it is the collaboration and not the sparseness of the representation that is the reason behind the effectiveness of SRC (and hence other approaches). This result is rather widely acclaimed as it provides grounds to relinquish the computationally expensive sparsity constraint over the representation without sacrificing the classification accuracy.

In this paper, we first extend the analysis of Zhang et al. [233] and, in contrast to the original claim, we show that sparseness of collaborative representation explicitly contributes to accurate classification, hence it should not be completely ignored for computational gains. Motivated by this intuition, we propose a Sparsity Augmented Collaborative Representation based Classification scheme (SA-CRC)\(^1\) that uses both dense and sparse collaborative representations to decide the class label of a test sample. SA-CRC computes the dense representation using the regularized least squares method and greedily approximates the sparse representation using the Orthogonal Matching Pursuit (OMP) [33]. OMP’s solution is used to augment the dense representation. Finally, the augmented representation is classified by capitalizing on its enriched discriminative properties. To that end, we propose an efficient classification method that avoids explicit computation of the reconstruction residuals for each class. We evaluate the proposed approach on two face databases [140], [77], one object category database [74] and a dataset for action recognition [172]. Extensive experiments with these public databases show that our approach is not only more accurate than the state-of-the-art collaborative representation based classification approaches, its classification time is also much lower than the approaches that ignore the sparsity altogether.

We organize this paper as follows. In Section 9.2, we formulate the problem and define the used terms. An overview of the relevant literature is provided in Section 9.3. Section 9.4 discusses the role of collaboration and sparsity in classification. We present the proposed approach in Section 9.5. Experimental evaluation of the approach is provided in Section 9.6. After a discussion on the parameter settings of our approach in Section 9.7, we conclude the paper in Section 9.8.

9.2 Problem Formulation

Let $\Phi \in \mathbb{R}^{m \times N}$ denote the training data from $C$ distinct classes, such that $\Phi = [\Phi_1, ..., \Phi_i, ..., \Phi_C]$. Each sub-matrix $\Phi_i \in \mathbb{R}^{m \times n_i}$ pertains to a single class and $\sum_{i=1}^{C} n_i = N$. The columns of $\Phi$ represent the training samples, that are the features extracted from images. Our goal is to develop an efficient multi-class classification scheme by collaboratively representing a test sample $y \in \mathbb{R}^m$ over the training data\(^2\). A test sample is considered to be a feature vector that can be linearly approximated by the training samples. That is, $y \approx \Phi \alpha$, where $\alpha \in \mathbb{R}^N$ is the Collaborative Representation (CR) vector of the test sample. We allow $\Phi$ to be a redundant set of basis vectors in $\mathbb{R}^m$. Furthermore, the subspaces spanned by the sub-matrices $\Phi_i \in \{1, ..., C\}$ are considered to be possibly overlapping, as this is often the case for the multi-class classification problems. Following the sparse representation literature [176], [212], we alternatively refer to $\Phi$ as the dictionary and to its columns as the dictionary atoms. Furthermore, we generally refer to the representation vector (e.g. $\alpha$) as representation, for brevity.

9.3 Related Work

Algorithm 8 presents the base-line scheme used by the popular approaches (e.g. [213], [64], [233], [182], [203], [221]) that exploit collaborative representation in multi-class classification. The algorithm performs three key steps of (1) optimizing $y$’s representation over a given dictionary, (2) computing class-specific reconstruction residuals $r_i(y), \forall i \in \{1, ..., C\}$ and (3) labeling $y$ using the computed residuals. In step (2), $\hat{\alpha} \in \mathbb{R}^{n_i}$ comprises the coefficients of $\alpha$ corresponding to the $r^{th}$ class only. Hence, in step (3), $y$ is assigned the label of the class that results in the smallest reconstruction residual. We can treat different existing approaches as special cases of the presented algorithm.

\(^2\)No explicit training of a machine learning algorithm is aimed, $\Phi$ is conventionally referred as the training data [213], [233].
Algorithm 8 CR-based Classification

**Input:** (a) Training data $\Phi$, with samples normalized to have unit $\ell_2$-norm. (b) Test sample $y$. (c) Regularization parameter $\lambda$.

1: **Optimization:** Solve
\[
\alpha = \min_{\alpha} ||y - \Phi \alpha||_2^2 + \lambda f(\alpha),
\]
where, $f(.)$ denotes a function and $||.||_p$ represents the $\ell_p$-norm of a vector.

2: **Residual computation:** Compute class-specific reconstruction residuals $r_i(y) = ||y - \Phi_i \hat{\alpha}_i||_2$, $\forall i \in \{1, ..., C\}$, where $\hat{\alpha}_i \in \mathbb{R}^{n_i}$ comprises the coefficients of $\alpha$ corresponding to the $i^{th}$ class.

3: **Labeling:** label$(y) = \min_i \{r_i(y)\}$.

**Output:** label$(y)$.

In SRC [213], $f(\alpha) = ||\alpha||_1$ in Eq. (9.1), which encourages the computed representation $\alpha$ to be sparse. In Superposed-SRC (SSRC), Deng et al. [64] modified the residual computation step of SRC. For SSRC, $\Phi$ consists of class centroids and sample-to-centroid differences. While computing the residuals, SSRC keeps the coefficients of $\alpha$ corresponding to the sample-to-centroid differences fixed in each $\hat{\alpha}_i$.

The CR-based classifier proposed by Zhang et al. [233] uses $f(\alpha) = ||\alpha||_2$ and solves Eq. (9.1) using the Regularized Least Squares (RLS) method, hence denoted as CRC-RLS. Shi et al. [182] used $\lambda = 0$ in Eq. (9.1) and solved it as the standard least squares problem for face recognition. Chi and Porikli [53] used a linear combination of a CR-based classifier and a nearest subspace classifier [122] for improved classification performance.

Collaborative representation is also commonly used by discriminative dictionary learning techniques, e.g. [203], [221]. Although such approaches learn a dictionary instead of directly using the training data as $\Phi$, explicit correspondence between the learned dictionary atoms and the class labels allows them to exploit the CR-based classification scheme. For instance, the Global Classifier (GC) used by Kong and Wang [203] is the same variant of Algorithm 8 that is used by SSRC [64]. The dictionary learned by the DL-COPAR algorithm [203] consists of COmmon atoms for all classes and PARticular atoms specific to each class. The particular atoms behave like class centroids whereas the common atoms act as centroid-to-sample differences in SSRC. Similarly, the GC used in the Fisher Discriminant Dictionary Learning (FDDL) [221] is a direct variant of CRC-RLS [233].

Another interesting direction of discriminative dictionary learning techniques,
9.4. Collaboration and Sparsity

It is clear from Section 9.3 that many popular approaches directly exploit collaborative representation \( \alpha \) in classification. Whereas sparse representation based approaches (e.g. [213], [64]) associate the discriminative power of \( \alpha \) to its sparseness, there is an equal evidence in favor of discriminative abilities of dense representations [53], [182], [233]. In fact, it is also advocated that sparsity of the representation may not even be relevant to classification [170], [182], [233]. Zhang et al. [233] boosted the popularity of this notion by corroborating their claim with an analysis of the working mechanism of SRC. In Section 9.4.1, we closely follow this analysis to explain the role of collaboration in CR-based classification. We extend this analysis on the same lines of reasoning in Section 9.4.2 to show that collaboration alone is not sufficient for accurate classification. Section 9.4.3 discusses how sparseness additionally helps in this regard.

9.4.1 Why collaboration works?

We write the subspace spanned by the columns of \( \Phi \) as a set \( \Psi \). This subspace is geometrically illustrated as a plane in Fig. 9.1. Since a test sample \( y \) is approximated by the columns of \( \Phi \), we can write the approximation error as \( \epsilon = y - \tilde{y} \), where \( \tilde{y} = \Phi \alpha \subset \Psi \). Let us represent the subspace spanned by the training data of the

\[ 3 \text{For } \Phi \in \mathbb{R}^{m \times N}, y \subset \Psi \text{ when } N \to \infty \text{ and } \epsilon \perp \tilde{\Psi}, \text{ where } \tilde{\Psi} \subset \Psi. \text{ In that case, we are concerned with } \tilde{\Psi} \text{ only, as } y \text{ is considered to be approximated with a small error of bounded energy, i.e. } ||\epsilon||_2 \leq \epsilon \text{. We exaggerate the error vector in figures for clarity.} \]
Figure 9.1: Geometric illustration of the working mechanism of collaborative representation based classification.

\( \bar{y} \) into two components, \( \xi_i \) and \( \bar{\xi}_i \) (illustrated in Fig. 9.1a) such that \( \xi_i \subset \Psi_i \) and \( \bar{\xi}_i \subset \Psi_i \), where \( \Psi_i = \bigcup_{j=1,j\neq i}^{C} \Psi_j \). Similarly, the total approximation error \( \epsilon \) can itself be considered as a component of \( \epsilon_i \), where \(|\epsilon_i|_2\) represents the class-specific reconstruction residual \( r_i(y) \), see step 2 of Algorithm 8.

9.4.2 Why collaboration alone is not sufficient?

In the following text, we refer to a vector \( \epsilon_i \) as class-specific error vector. We present Lemma 4 regarding the underlying geometry of the class-specific error vectors involved in CR-based classification:

**Lemma 4.** For \( i, j, k \in \{1, ..., C\} \), where \( i \neq j \neq k \), the following holds: \( \exists \epsilon_i, \epsilon_j \) such that \(|\epsilon_i|_2 = |\epsilon_j|_2\), while \( \nexists \epsilon_k \) such that \(|\epsilon_k|_2 < |\epsilon_i|_2\).

**Proof:** For our problem, the following holds under the law of sines, which can be
verified from Fig. 9.1a:

\[ \frac{||\tilde{y}||_2}{\sin(\gamma)} = \frac{||\xi_i||_2}{\sin(\beta)}. \]  

(9.2)

Also, \[ ||\epsilon_i||_2^2 = ||\epsilon||_2^2 + ||\xi_i||_2^2 \] because \( \Psi \perp \epsilon \). From Eq. (9.2),

\[ ||\epsilon_i||_2^2 = ||\epsilon||_2^2 + \left( \frac{\sin(\beta)}{\sin(\gamma)} \right)^2 ||\tilde{y}||_2^2. \]  

(9.3)

Since \( ||\epsilon||_2^2 \) and \( ||\tilde{y}||_2^2 \) become constants once \( y \) is projected onto \( \Psi \), the condition that \( \nexists \epsilon_k \) s.t. \( ||\epsilon_k||_2 < ||\epsilon_i||_2 \) holds when \( \left( \frac{\sin(\beta)}{\sin(\gamma)} \right)^2 \) is the minimum. However, for \( \beta, \gamma \in [0, 2\pi] \) there is no unique minima for the given squared ratio. Hence, it is possible that \( \exists \epsilon_i, \epsilon_j \) s.t. \( ||\epsilon_i||_2 = ||\epsilon_j||_2 \), while \( \nexists \epsilon_k \) s.t. \( ||\epsilon_k||_2 < ||\epsilon_i||_2 \).

Lemma 4, shows the possibility of existence of multiple class-specific error vectors with equal lengths when the length is minimized over the class labels. Figure 9.1b illustrates this possibility by drawing a circle of radius \( ||\xi_i||_2 \) around point \( \tilde{o} \) on \( \Psi \). Any vector starting from a point on this circle (e.g. \( p, q \)) and ending at \( z \) will have the same length. For the labeling criterion of CR-based classification scheme, collaboration of the representation alone is not sufficient to indicate the best vector among these possible vectors. From Lemma 4, it is also evident that the double-check with \( \beta \) and \( \gamma \) mentioned by Zhang et al. [233] is essentially a single-check on the squared ratio of the sines of the angles. Thus, CR-based classification without considering sparsity may not be as robust and effective as previously thought.

### 9.4.3 How sparseness helps?

The above mentioned issue is inherent to CR-based classification scheme, with its roots in the redundancy in \( \Phi \). Simply computing a unique approximation of the representation, such as in CRC-RLS [233], does not resolve the issue because Lemma 4 still holds for the labeling step in Algorithm 8. To truly address the problem, a collaborative representation must be infused with additional information that finally results in using a suitable class-specific error vector in the labeling step. Sparsity constraint over the representation serves this purpose in CR-based classification.

To support our argument, in Fig. 9.2, we geometrically illustrate the two jointly exhaustive situations that can occur when two class-specific error vectors \( \epsilon_i \) and \( \epsilon_j \) have equal lengths, namely (a) \( \xi_i \neq \xi_j \) and (b) \( \xi_i = \xi_j \). In the figure, we denote \( \xi_i \) by \( a \) and \( \xi_j \) by \( b \) and show these vectors only by their components to avoid cluttering. In Fig. 9.2a, \( a \neq b \) but \( ||\epsilon_i||_2 = ||\epsilon_j||_2 \). In Fig. 9.2b, \( a = b = \tilde{op} \) and
Chapter 9. Sparsity Augmented CRC

Figure 9.2: Geometric illustration of the jointly exhaustive cases when $\exists \epsilon_i, \epsilon_j$ such that $||\epsilon_i||_2 = ||\epsilon_j||_2$: (a) $\xi_i \neq \xi_j$. (b) $\xi_i = \xi_j$. Here, $\xi_i = a$ and $\xi_j = b$ and the vectors are only displayed in terms of their components.

![Geometric illustration](image)

$||\epsilon_i||_2 = ||\epsilon_j||_2$. Although the class-specific residuals are equal in both cases, $\xi_i$ and $\xi_j$ can be distinguished based on their components. Intuitively, $i$ (not $j$) represents the correct class of the test sample because $\xi_i$ requires lesser number of components to produce the smallest class-specific residual. Fewer components of $\xi_i$ implicates a sparser $\alpha$. Hence, the sparsity constraint results in using a better class-specific error vector in the labeling step. Incidentally, the best performance of CR-based classification can be achieved by guaranteeing the representation to be the sparsest possible.

9.5 Proposed Approach

Computing the sparsest possible representation is generally NP-hard [148]. SRC [213] uses the $\ell_1$-norm constraint to compute an approximate sparse representation, but the approach remains computationally expensive. On the other hand, computing a dense representation, such as in CRC-RLS [233], resolves the computational issues but it does not offer the advantages of sparsity. In the proposed classification scheme, we augment a dense representation with a greedily obtained approximate sparse representation. This augmentation enables accurate classification while keeping the approach computationally efficient.

Algorithm 9 presents the proposed scheme. In the first step, the algorithm optimizes two collaborative representations, i.e. $\hat{\alpha}$ and $\tilde{\alpha}$. The dense representation $\hat{\alpha}$ is computed using the regularized least squares method, whereas the sparse representation $\tilde{\alpha}$ is obtained by solving Eq. (9.4) using the Orthogonal Matching Pursuit (OMP) algorithm [33]. OMP iteratively selects $k$ dictionary atoms to represent $y$,
9.5. Proposed Approach

Algorithm 9 Sparsity Augmented CR-based Classification

**Input:** (a) Training data $\Phi$, with samples normalized in $\ell_2$-norm. (b) Test sample $y$. (c) Regularization parameter $\lambda$. (d) Sparsity threshold $k$. (e) Label matrix $L$.

1: **Optimization:**
   a) Compute $\hat{\alpha} = Py$, where $P = (\Phi^T \Phi + \lambda I_N)^{-1} \Phi^T$.
   b) Solve the following with greedy pursuit:

   $$\hat{\alpha} = \min_{\alpha} ||y - \Phi \alpha||_2, \ s.t. \ ||\alpha||_0 \leq k,$$

   where, $||.||_0$ denotes the $\ell_0$-pseudo norm.

2: **Augmentation:** Compute

   $$\tilde{\alpha} = \frac{\hat{\alpha} + \hat{\alpha}}{||\hat{\alpha} + \hat{\alpha}||_2}.$$  \hspace{1cm}(9.5)

3: **Labeling:** $\text{label}(y) = \arg \max_i \{ q_i \}$, where $q_i$ denotes the $i^{th}$ coefficient of $q = L\tilde{\alpha}$.

**Output:** $\text{label}(y)$.

hence, $\hat{\alpha}$ has at most $k$ non-zero coefficients, where $k$ (the sparsity threshold) is determined by cross-validation. In each iteration, OMP chooses a dictionary atom by maximizing its correlation with an error vector. The error vector is computed as the difference between $y$ and its orthogonal projection onto the subspace spanned by the already chosen atoms. For initialization, $y$ itself is considered as the error.

As shown in step 2 of Algorithm 9, we add the sparse representation $\hat{\alpha}$ to $\hat{\alpha}$ and normalize the resulting vector to compute the augmented representation $\tilde{\alpha}$. Despite being simple, this procedure greatly improves the discriminative abilities of the representation. We defer the discussion on the discriminative properties of $\tilde{\alpha}$ to the upcoming paragraphs. These properties are exploited in step (3) of the algorithm to efficiently compute the label of the test sample $y$. The labeling step uses a binary matrix $L \in \mathbb{R}^{C \times N}$, that is provided as an input to the algorithm. For the $i^{th}$ class, $L$ contains $n_i$ non-zero elements in its $i^{th}$ row, at the indices corresponding to the columns of $\Phi_i$. Thus, the $i^{th}$ coefficient of $q = L\tilde{\alpha}$ represents the sum of $\tilde{\alpha}$’s coefficients corresponding to $\Phi_i$. The label of the test sample is decided by maximizing the coefficients of $q$. Empirical evidence for efficient and accurate classification using the proposed scheme is provided in Sections 9.6. Below, we analyze the reasons behind the improved performance of the approach.
Chapter 9. Sparsity Augmented CRC

For analysis, let us distribute the coefficient indices of a collaborative representation $\mathbf{\alpha}$ into two disjoint sets: $\mathcal{A}_H = \{i : \Xi_i > \delta \}$ and $\mathcal{A}_L = \{j : \Xi_j \leq \delta \}$, where $\delta \in \mathbb{R}^+$ and $\Xi_n = \frac{\alpha_n^2}{||\alpha||^2_2}$ with $\alpha_n \in \mathbb{R}$ denoting the $n^{th}$ coefficient of $\mathbf{\alpha}$. The value $\Xi_n$ represents the energy in the $n^{th}$ coefficient, such that $\sum_{n=1}^{N} \Xi_n = 1$. If $\delta = 0$, $\mathcal{A}_H$ contains the indices of non-zero coefficients of $\mathbf{\alpha}$, whereas $\mathcal{A}_L$ comprises the indices of zero coefficients. Thus, the cardinality of the set $\mathcal{A}_H$, i.e. $|\mathcal{A}_H|$, defines the sparsity level of $\mathbf{\alpha}$. This remains true for $0 \leq \delta < \min_i \Xi_i$. Let $\mathbf{\alpha}^*$ denote the sparsest possible representation of $\mathbf{y}$ over $\Phi$. We write the aforementioned sets for $\mathbf{\alpha}^*$ as $\mathcal{A}^*_H$ and $\mathcal{A}^*_L$. Furthermore, for any $\mathbf{\alpha}$, let us now fix $\delta = \left(\frac{\alpha_{\min}^*}{||\alpha^*||_2}\right)^2 - \varepsilon$, where $\alpha_{\min}^*$ denotes the lowest energy coefficient of $\mathbf{\alpha}^*$. Hence, $|\mathcal{A}_H|$ now counts the number of coefficients of $\mathbf{\alpha}$, each having at least the energy possessed by $\alpha_{\min}^*$. Therefore, henceforth, we refer to $|\mathcal{A}_H|$ as the effective sparsity of the representation.

From Section 9.4, we know that $\mathbf{\alpha}^*$ is discriminative due to its sparsity. In practice, a representation $\hat{\mathbf{\alpha}}$ is equally effective for classification if $|\mathcal{A}_{\hat{\mathbf{\alpha}}}^*_H| \approx |\mathcal{A}_H^*|$ and the coefficients indexed in $\mathcal{A}_{\hat{\mathbf{\alpha}}}^*_H$ are discriminative. For a dense representation $\mathbf{\hat{\alpha}}$, $|\mathcal{A}_{\mathbf{\hat{\alpha}}}^*| \approx N \gg |\mathcal{A}_H^*|$. Nevertheless, the representation is globally optimal. On the other hand, $|\mathcal{A}_{\hat{\mathbf{\alpha}}}^*_H| \approx k \ll N$ for the sparse representation $\mathbf{\hat{\alpha}}$, but the representation is only locally optimal. However, $\hat{\mathbf{\alpha}}$ generally contains large positive coefficients at the indices corresponding to the correct class. For the other classes, most of the coefficients are either negative or have small positive values. This happens because OMP greedily assigns large values to the coefficients of $\hat{\mathbf{\alpha}}$ corresponding to the dictionary atoms that correlate more to $\mathbf{y}$, whereas $\mathbf{y}$ generally has a strong positive correlation with the samples of its own class. Thus, adding $\hat{\mathbf{\alpha}}$ to $\mathbf{\alpha}$ amplifies the coefficients of the correct class in the globally optimal solution. Figure 9.3 illustrate this phenomenon using an actual example of face recognition. In the figure, the coefficients of $\hat{\mathbf{\alpha}}$ are consistently positive and have relatively large values for the correct class. This finally results in dominant positive coefficients of $\hat{\mathbf{\alpha}}$ for the correct class. For this example, CRC-RLS [233] is not able to identify the correct label of $\mathbf{y}$ despite optimized parameter settings, whereas the proposed approach classifies $\mathbf{y}$ correctly.

Notice that, the augmentation in Eq. 9.5 also results in $|\mathcal{A}_H^*| \ll |\mathcal{A}_{\hat{\mathbf{\alpha}}}^*_H|$, because the procedure reduces the relative energy in the un-amplified coefficients of $\hat{\mathbf{\alpha}}$. To illustrate the difference between the effective sparsity levels of the dense and the

---

4We can safely ignore $\mathcal{A}_L^*$ in this argument because the coefficients indexed in $\mathcal{A}_L^*$ can be explicitly forced to zero, once $\mathcal{A}_{\hat{\mathbf{\alpha}}}^*_H$ is known.
9.5. Proposed Approach

Figure 9.3: Comparison of test sample representations from the Extended YaleB database [77]: The sparse representation consistently shows large positive values at the coefficients corresponding to the correct class. This results in correct classification by SA-CRC that uses the augmented representation to predict the class label. By using the dense representation only, CRC-RLS [233] predicts incorrect class label despite optimized parameter values. For better visibility, only the first 900 coefficients of the representation vectors are shown out of 1216.

augmented representations, we plot the effective sparsity of the representations as a function of $\delta$ in Fig. 9.4. The plot is for actual face recognition task using Extended YaleB database [77]. The curve for the augmented representation remains significantly lower than the curve for the dense representation. Moreover, for $\delta > 3 \times 10^{-4}$, $\hat{\alpha}$ is effectively almost as sparse as $\tilde{\alpha}$.

Considering the definition of effective sparsity, ideally, the coefficients of $\hat{\alpha}$ indexed in $\mathcal{A}_C^\circ$ must be forced to zero before using the representation for classification. However, since $\delta$ is unknown, identifying the exact $\mathcal{A}_C^\circ$ remains NP-hard. To resolve this issue, we design the labeling criterion that largely remains insensitive to the coefficients indexed in $\mathcal{A}_C^\circ$. That is, instead of deciding the class label of a test sample based on the fidelity of its reconstruction, we directly integrate the coefficients of $\hat{\alpha}$ for each class separately. The largest integrated value indicates the correct class label. Due to the dominance of large values of the coefficients of the correct class in $\hat{\alpha}$, $\mathcal{A}_C^\circ$ is not able to strongly influence the classification results. More precisely, our classification result remains as reliable as that obtained using an accurate representation with sparsity level $|\mathcal{A}_F^\circ|$, under the mild worst-case condition $\sum_a - \sum_b > 2\sqrt{\delta}(n_b - n_a)$. Here, $\sum_a$ and $\sum_b$ denote the largest and the second
largest integrated values of the coefficients, respectively, and $n_b$ and $n_a$ are the number of coefficients in $\hat{\alpha}$ contributing to $\sum_b$ and $\sum_a$ respectively, such that, each coefficient has energy less than $\delta$. To exemplify, in Fig. 9.4, the classification results are as accurate as possible with sparsity level 21, unless $\sum_a - \sum_b \leq 0.04 \times (n_b - n_a)$. Typically, $\sum_a - \sum_b \in [0.1, 0.3]$, whereas $n_b \approx n_a$. Since our labeling criterion does not need to compute reconstruction residuals for each class, we directly use the matrix $L$ in step (3) of Algorithm 9. The matrix multiplication $L\hat{\alpha}$ simultaneously integrates the coefficients for each class. Computationally, this makes our labeling step very efficient.

9.6 Experiments

We evaluated the proposed approach on two face databases: AR database [140] and Extended YaleB [77], an object category database: Caltech-101 [74] and an action dataset: UCF sports actions [172]. These datasets are commonly used to benchmark the approaches that use collaborative representation for classification. We compare the performance of our approach to SRC [213], CRC-RLS [233], LC-KSVD [104], D-KSVD [234], FDDL [221] and DL-COPAR [203]. Unless mentioned otherwise, we performed our own experiments using the same training and testing partitions for all the approaches including the proposed approach. We carefully optimized the parameter values of the approaches using cross validation. For the existing techniques, these values are generally the same as those reported in the original works. However, for some cases, we used different values to favors these approaches. We explicitly mention these differences. For the dictionary learning approaches, the dictionaries are learned using the same training data that is directly used by SRC, CRC-RLS and the proposed approach.
9.6. Experiments

We used the author-provided codes for CRC-RLS, LC-KSVD, FDDL and DL-COPAR. For SRC, we used the SPAMS toolbox [135] to solve the $\ell_1$-norm minimization problem. For D-KSVD, we modified the public code of LC-KSVD [104]. In all the experiments, the proposed approach uses the implementation of OMP made public by Elad et al. [177]. The same implementation is used by LC-KSVD and D-KSVD. The proposed approach uses the sparsity threshold $k = 50$ for all the datasets. The regularization parameter $\lambda$ is set to 0.003 for the face databases, 1.0 for the object database and 0.01 for the action database. Experiments have been performed using an Intel processor at 3.4 GHz with 8 GB RAM.

9.6.1 AR Database

The AR database [140] consists of over 4,000 face images of 126 subjects. For each subject, 26 images are taken during two different sessions with large variations in terms of facial disguise, illumination and expressions. Please see Fig. 10.5b (Section 10.5.1) for illustrations. For our experiments, a $165 \times 120$ face image was projected onto a 540-dimensional vector using a random projection matrix. Thus, the used samples are the Random-Face features [213]. We followed a common experimental protocol by selecting a subset of 2,600 images of 50 male and 50 female subjects from the database. For each subject, 20 random images were chosen to create the training data and the remaining images were used for testing.

In Table 9.1, we summarize the results on the AR database. The reported accuracies are the means (and the standard deviations) of ten experiments. We also report the average time taken by each approach to classify a single test sample. For the parameter values of DL-COPAR, we followed the face recognition parameter settings in [203], which uses 15 atoms per class to represent class-specific data and 5 atoms to represent the commonalities. The Local Classifier [203] resulted in the best accuracy for DL-COPAR. For LC-KSVD [104] and D-KSVD [234] we set the sparsity threshold to 50 and the dictionary size to 1510 atoms for improved results. These values are different from the original works because these were found to give the best accuracies. For FDDL, we used the same parameter settings as [221] and the Global Classifier resulted in the best performance. For SRC [213], we set the error tolerance $\varepsilon = 0.05$, as in the original work. For CRC-RLS [233], the regularization parameter $\lambda$ is set to 0.003. This value is computed using the formula provided for $\lambda$ for the face databases in [233]. Our cross-validation verified that this value results in the best performance of CRC-RLS.

Table 9.1 shows that the best results are achieved by the proposed approach,
Table 9.1: Recognition accuracies on the AR database [140] using Random-Face features. The reported average time (milli-seconds) is for a single test sample.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>DL-COPAR [203]</td>
<td>93.33 ± 1.69</td>
<td>40.01</td>
</tr>
<tr>
<td>LC-KSVD [104]</td>
<td>95.20 ± 1.22</td>
<td>1.56</td>
</tr>
<tr>
<td>D-KSVD [234]</td>
<td>95.41 ± 1.43</td>
<td>1.54</td>
</tr>
<tr>
<td>FDDL [221]</td>
<td>96.24 ± 1.01</td>
<td>51.23</td>
</tr>
<tr>
<td>SRC [213]</td>
<td>96.51 ± 1.36</td>
<td>69.91</td>
</tr>
<tr>
<td>CRC-RLS[233]</td>
<td>97.65 ± 0.67</td>
<td>4.46</td>
</tr>
<tr>
<td>SA-CRC (only RLS)</td>
<td>97.13 ± 0.74</td>
<td>0.07</td>
</tr>
<tr>
<td>SA-CRC (only OMP)</td>
<td>97.25 ± 0.43</td>
<td>2.00</td>
</tr>
<tr>
<td>SA-CRC (proposed)</td>
<td>98.29 ± 0.46</td>
<td>2.13</td>
</tr>
</tbody>
</table>

Table 9.2: Performance gain with SA-CRC in dictionary learning based classification. The average time (in milli-seconds) is for classifying a single test sample.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-SVD [2] + Lin. Classifier</td>
<td>94.06 ± 1.03</td>
<td>1.56</td>
</tr>
<tr>
<td>K-SVD $\Phi \rightarrow$ SA-CRC</td>
<td>95.65 ± 0.66</td>
<td>1.61</td>
</tr>
<tr>
<td>ODL [135] + Lin. Classifier</td>
<td>94.60 ± 0.78</td>
<td>1.59</td>
</tr>
<tr>
<td>ODL $\Phi \rightarrow$ SA-CRC</td>
<td>95.33 ± 0.68</td>
<td>1.62</td>
</tr>
<tr>
<td>LC-KSVD [104]</td>
<td>95.31 ± 1.06</td>
<td>1.55</td>
</tr>
<tr>
<td>LC-KSVD $\Phi, L \rightarrow$ SA-CRC</td>
<td>96.44 ± 0.99</td>
<td>1.61</td>
</tr>
</tbody>
</table>

i.e. SA-CRC. We have also shown the results of our approach when we use only the Regularized Least Squares (RLS) or OMP in Algorithm 9. It is clear that using the augmented vector is better than using any of the two representation vectors alone. Notice that, due to the efficient classification criterion, our approach is much faster than CRC-RLS even when both OMP and RLS are used. The dictionaries used by LC-KSVD and D-KSVD are smaller in size as compared to the one used by SA-CRC, which results in slight computational advantage for these approaches. Nevertheless, accuracies of these approaches are much lower than SA-CRC.

In Table 9.2, we demonstrate the potential of SA-CRC for improving the performance of dictionary learning based multi-class classification approaches. The results are the mean values computed over ten experiments. We obtained the results in the
first row as follows. First, K-SVD [2] is used to learn a dictionary containing 15 atoms per class. The sparse codes of the training data over the dictionary are used to compute a linear classifier, following [104]. A test sample is classified by first sparse coding it over the dictionary and then classifying its codes using the classifier. In the second row, we feed the same dictionary to SA-CRC as input $\Phi$ instead (the test data remained the same). We also repeated the above procedure using the Online Dictionary Learning (ODL) approach [135] in place of K-SVD. The corresponding results are also reported. We can see a clear gain in the classification accuracies using SA-CRC in both cases. The table also compares the classification performance of LC-KSVD [104] with its enhancement using SA-CRC. For the enhancement, we replaced the classification stage of LC-KSVD with SA-CRC. That is, the dictionary and the weight matrix learned by LC-KSVD are directly used as SA-CRC’s inputs $\Phi$ and $L$, respectively. There is a clear improvement in the classification performance of LC-KSVD after this modification. The performance of other discriminative dictionary learning approaches can also be improved using SA-CRC. Results in Table 9.1 and 9.2 demonstrate the potential of sparsity augmented collaborative representation for improved CR-based classification across the board.

### 9.6.2 Extended YaleB

The Extended YaleB face database [77] comprises 2,414 images of 38 subjects. Each subject has about 64 samples acquired under varying illumination conditions with different expressions. Please see Fig. 10.5a (Section 10.5.1) for examples. For this database, 192 × 168 cropped images were projected onto a 504-dimensional vector to obtain the Random-Face features. For evaluation, we used a common experimental setting, where half of the available features of each subject were used as the training data and the remaining half were used in testing.

In Table 9.3, we show the results on Extended YaleB. For D-KSVD and LC-KSVD we used 600 dictionary atoms as they gave the best accuracies. The remaining parameters of these algorithms were set to the original values reported in [104]. We set the regularization parameter of CRC-RLS to 0.002 for this database, as guided by [233] and dictated by cross-validation. For the remaining approaches, the parameter values reported in Section 9.6.1 also resulted in their best performances for this database, hence they were kept the same. Again, SA-CRC is able to outperform the existing techniques. Although SA-CRC attains only a slight advantage over CRC-RLS in terms of accuracy for this dataset, it is able to classify a test sample almost twice as fast as CRC-RLS due to the proposed classification criterion.
Table 9.3: Recognition accuracies with Random-Face features on the Extended YaleB [77]. The average time (milli-seconds) is for classifying a single sample.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-KSVD [234]</td>
<td>94.71 ± 0.45</td>
<td>0.41</td>
</tr>
<tr>
<td>DL-COPAR [203]</td>
<td>94.87 ± 0.55</td>
<td>31.75</td>
</tr>
<tr>
<td>LC-KSVD [104]</td>
<td>95.38 ± 0.64</td>
<td>0.42</td>
</tr>
<tr>
<td>FDDL [221]</td>
<td>96.19 ± 0.71</td>
<td>58.19</td>
</tr>
<tr>
<td>SRC [213]</td>
<td>97.06 ± 0.41</td>
<td>68.12</td>
</tr>
<tr>
<td>CRC-RLS [233]</td>
<td>97.81 ± 0.44</td>
<td>2.41</td>
</tr>
<tr>
<td>SA-CRC</td>
<td><strong>98.32 ± 0.43</strong></td>
<td>1.23</td>
</tr>
</tbody>
</table>

9.6.3 Caltech-101

Caltech-101 database [74] contains 9,144 images from 101 object classes and one class of background images. The classes include diverse categories of object (e.g. trees, minarets, signs) with significant shape variation within a category. Fig. 10.6 illustrates this variation in Section 10.5.3. For each class, the number of available images vary between 31 and 800. In our experiments, the used image feature descriptors were obtained by the following procedure. First, the SIFT descriptors [126] were extracted from 16 × 16 patches. Based on these descriptors, spatial pyramid features [219] were extracted with 1 × 1, 2 × 2 and 4 × 4 grids. For extracting these features, the codebook was trained using k-means, where \( k = 1024 \). Finally, the dimension of a feature was reduced to 3,000 using PCA. Following a common experimental setting, we created 5 sets of train and test data with the extracted features. The sets comprised 10, 15, 20, 25 and 30 training samples per class, whereas the remaining samples were used as the test data in each case. We repeated our experiments ten times, every time selecting the training and testing data randomly.

Table 9.4 shows the mean classification accuracies for our experiments. We used the error tolerance of \( 10^{-6} \) for SRC, which gave the best results. The regularization parameter \( \lambda = 1.0 \) for CRC-RLS. The same value of the regularization parameter is used by SA-CRC to solve the RLS problem. For FDDL, we used the parameter settings of the object categorization experiments in [221]. DL-COPAR and LC-KSVD use the same settings as in the original works for the same database. These settings also resulted in their best performance for our data. For D-KSVD, the settings used by [104] showed the best results.

It is clear from Table 9.4 that SA-CRC consistently outperforms the existing
9.6. Experiments

Table 9.4: Classification accuracies (%) on the Caltech-101 dataset [74] using the spatial pyramid features.

<table>
<thead>
<tr>
<th>Training samples</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRC [213]</td>
<td>57.8</td>
<td>63.3</td>
<td>67.2</td>
<td>69.2</td>
<td>71.8</td>
</tr>
<tr>
<td>CRC-RLS [233]</td>
<td>59.4</td>
<td>64.8</td>
<td>68.0</td>
<td>69.3</td>
<td>71.8</td>
</tr>
<tr>
<td>DL-COPAR [203]</td>
<td>58.4</td>
<td>65.1</td>
<td>69.3</td>
<td>71.1</td>
<td>72.5</td>
</tr>
<tr>
<td>FDDL [221]</td>
<td>59.7</td>
<td>66.6</td>
<td>69.1</td>
<td>71.3</td>
<td>72.9</td>
</tr>
<tr>
<td>D-KSVD [234]</td>
<td>60.7</td>
<td>66.3</td>
<td>69.6</td>
<td>71.0</td>
<td>73.1</td>
</tr>
<tr>
<td>LC-KSVD [104]</td>
<td>62.9</td>
<td>67.3</td>
<td>70.3</td>
<td>72.6</td>
<td>73.4</td>
</tr>
<tr>
<td>SA-CRC</td>
<td>63.2</td>
<td>68.2</td>
<td>71.9</td>
<td>73.6</td>
<td>76.1</td>
</tr>
</tbody>
</table>

Table 9.5: Computation time (in seconds) for classification on Caltech-101 [74].

<table>
<thead>
<tr>
<th>Method</th>
<th>Time</th>
<th>Method</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-KSVD [234]</td>
<td>19.80</td>
<td>SA-CRC</td>
<td>21.43</td>
</tr>
<tr>
<td>LC-KSVD [104]</td>
<td>19.91</td>
<td>CRC-RLS [233]</td>
<td>130.41</td>
</tr>
</tbody>
</table>

approaches. In Table 9.5, we also report the classification time (for the complete test data) of the four most efficient approaches. The time is computed when 30 samples were used for training and the rest were used for testing. We can see that SA-CRC is more than six times faster than CRC-RLS and its timings are comparable to those of the efficient discriminative dictionary learning approaches. Note that, D-KSVD and LC-KSVD also required around 90 minutes of training.

9.6.4 UCF Sports Actions

The UCF Sports Action dataset [172] contains video sequences collected from different broadcast sports channels. The videos are from 10 categories of sports actions (e.g. diving, lifting, running). Fig. 10.8 shows eight representative images from the database for illustration in Section 10.5.5. For this dataset, we used the action bank features made public by Sadanand and Corso [179] (http://www.cse.buffalo.edu/~jcorso/r/actionbank/). A common evaluation protocol was followed in our experiments, where a fivefold cross validation was performed using four folds for training and the remaining one for testing. The results in Table 9.6 are the average accuracies of the five experiments. The reported accuracies of Sadanand [179], DL-COPAR and FDDL are taken directly from [220], where the same experimental protocol has been followed. Our parameter optimization for
Table 9.6: Classification accuracies (%) on UCF Sports Action dataset [172] using the action bank features.

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc.</th>
<th>Method</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sadanand [179]</td>
<td>90.7</td>
<td>FDDL [221]</td>
<td>93.6</td>
</tr>
<tr>
<td>DL-COPAR [203]</td>
<td>90.7</td>
<td>LC-KSVD [104]</td>
<td>94.2</td>
</tr>
<tr>
<td>D-KSVD [234]</td>
<td>93.4</td>
<td>CRC-RLS [233]</td>
<td>94.4</td>
</tr>
<tr>
<td>SRC [213]</td>
<td>93.5</td>
<td>SA-CRC</td>
<td>95.7</td>
</tr>
</tbody>
</table>

FDDL and DL-COPAR could not achieve these accuracies. For the remaining approaches the results are reported on the same folds using the optimized parameter values. For SRC, the error tolerance was set to $10^{-6}$ and 50 dictionary atoms were used for LC-KSVD and D-KSVD. The same number of atoms were used by Jiang et al. [104]. We used $\lambda = 0.01$ for both CRC-RLS and SA-CRC, which resulted in their best performance. For all the five experiments combined, the classification time was 0.04 seconds for SA-CRC and 0.31 seconds for CRC-RLS.

9.7 Discussion

Our approach requires a regularization parameter $\lambda$ and sparsity threshold $k$ as the input parameters for a given pair of $\Phi$ and its label matrix $L$. In our experiments, we optimized the values of these parameters by cross-validation using the following systematic procedure. First, $\lambda$ was optimized by executing Algorithm 9 without step 1(b) and considering $\hat{\alpha}$ to be a zero vector in Eq. 9.5. Then $k$ was optimized by fixing $\lambda$ to the optimized value and executing the complete Algorithm. The parameters were further fine-tuned to nearby values when doing so yielded better performance.

To show the behavior of SA-CRC for different parameter values, in Fig. 9.5, we plot the classification accuracy of SA-CRC as a function of $\lambda$ and $k$ by fixing one parameter and varying the other. We also include results of CRC-RLS [233] for comparison. Plots in Fig. 9.5a, are for AR database [140] where we followed the experimental protocol of [233]. In the first plot (from left), we fixed $k$ to 50 and varied $\lambda$. Clearly, SA-CRC consistently outperforms CRC-RLS and the results are less sensitive to the values of $\lambda$ once $k$ is fixed to an optimized value. In the second plot, we used $\lambda = 0.003$ for both SA-CRC and CRC-RLS and varied $k$ for SA-CRC. Again, for $k > 20$, SA-CRC consistently outperforms CRC-RLS. Qualitatively speaking, Fig. 9.5a shows a typical relationship between the performance of CRC-
9.8. Conclusion

In contrast to a popular existing notion, we showed that sparsity of a Collaborative Representation (CR) plays an explicit role in accurate CR-based classification, hence it should not be completely ignored for computational gains. Inspired by this result, we proposed a Sparsity Augmented Collaborative Representation based Classification scheme (SA-CRC) that augments a dense collaborative representation with an efficiently computed sparse representation. The resulting representation is classified using an efficient criterion. Extensive experiments for face, action and object classification establish the effectiveness of SA-CRC in terms of accuracy as well as computational efficiency.

Figure 9.5: Accuracy as a function of parameters: One parameter is fixed for SA-CRC while the other is varied. Both CRC-RLS [233] and SA-CRC always use the same values of $\lambda$. (a) $k$ is fixed to 50 in the first plot (from left) and $\lambda$ is fixed to 0.003 in the second. (b) $k$ is fixed to 50 and $\lambda$ to 1, respectively. Notice, that the used fixed values of $\lambda$ are the optimal values for CRC-RLS. However, the proposed approach consistently outperforms CRC-RLS, even for sub-optimal values of $\lambda$.

RLS and SA-CRC that was observed in our experiments on face databases.

In Fig. 9.5b, we repeated the same experiment for the object dataset, Caltech-101 [74]. To fix the parameter values, we used $k = 50$ and $\lambda = 1$. For this experiment, we used five samples per class for training and the remaining for testing. Again, the plots consistently favor SA-CRC in comparison to CRC-RLS. In our experiments, a qualitatively similar behavior was observed for all the train/test partitions used in Table 9.4.

9.8 Conclusion

In contrast to a popular existing notion, we showed that sparsity of a Collaborative Representation (CR) plays an explicit role in accurate CR-based classification, hence it should not be completely ignored for computational gains. Inspired by this result, we proposed a Sparsity Augmented Collaborative Representation based Classification scheme (SA-CRC) that augments a dense collaborative representation with an efficiently computed sparse representation. The resulting representation is classified using an efficient criterion. Extensive experiments for face, action and object classification establish the effectiveness of SA-CRC in terms of accuracy as well as computational efficiency.
CHAPTER 10

Discriminative Bayesian Dictionary Learning for Classification

Abstract

We propose a Bayesian approach to learn discriminative dictionaries for sparse representation of data. The proposed approach infers probability distributions over the atoms of a discriminative dictionary using a finite approximation of Beta Process. It also computes sets of Bernoulli distributions that associate class labels to the learned dictionary atoms. This association signifies the selection probabilities of the dictionary atoms in the expansion of class-specific data. Furthermore, the non-parametric character of the proposed approach allows it to infer the correct size of the dictionary. We exploit the aforementioned Bernoulli distributions in separately learning a linear classifier. The classifier uses the same hierarchical Bayesian model as the dictionary, which we present along the analytical inference solution for Gibbs sampling. For classification, a test instance is first sparsely encoded over the learned dictionary and the codes are fed to the classifier. We performed experiments for face and action recognition; and object and scene-category classification using five public datasets and compared the results with state-of-the-art discriminative sparse representation approaches. Experiments show that the proposed Bayesian approach consistently outperforms the existing approaches.

Keywords: Bayesian sparse representation, Discriminative dictionary learning, Supervised learning, Classification.

10.1 Introduction

Sparse representation encodes a signal as a sparse linear combination of redundant basis vectors. With its inspirational roots in human vision system [155], [156], this technique has been successfully employed in image restoration [71], [134], [8], compressive sensing [36], [67] and morphological component analysis [29]. More recently, sparse representation based approaches have also shown promising results in face recognition and gender classification [104], [213], [221], [220], [225], [223], [222], texture and handwritten digit classification [191], [218], [171], [132], natural image
and object classification [104], [203], [219] and human action recognition [165], [84], [204], [38]. The success of these approaches comes from the fact that a sample from a class can generally be well represented as a sparse linear combination of the other samples from the same class, in a lower dimensional manifold [213].

For classification, a discriminative sparse representation approach first encodes the test instance over a dictionary, i.e. a redundant set of basis vectors, known as atoms. Therefore, an effective dictionary is critical for the performance of such approaches. It is possible to use an off-the-shelf basis (e.g. fast Fourier transform [56] or wavelets [139]) as a generic dictionary to represent data from different domains/classes. However, research in the last decade ([2], [104], [221],[203], [71], [32], [238], [159]) has provided strong evidence in favor of learning dictionaries using the domain/class-specific training data, especially for classification and recognition tasks [221], where class label information of the training data can be exploited in the supervised learning of a dictionary.

Whereas unsupervised dictionary learning approaches (e.g. K-SVD [2], Method of Optimal Directions [72]) aim at learning faithful signal representations, supervised sparse representation additionally strives for making the dictionaries discriminative. For instance, in Sparse Representation based Classification (SRC) scheme, Wright et al. [213] constructed a discriminative dictionary by directly using the training data as the dictionary atoms. With each atom associated to a particular class, the query is assigned the label of the class whose associated atoms maximally contribute to the sparse representation of the query. Impressive results have been achieved for recognition and classification using SRC, however, the computational complexity of this technique becomes prohibitive for large training data. This has motivated considerable research on learning discriminative dictionaries that would allow sparse representation based classification with much lower computational cost.

In order to learn a discriminative dictionary, existing approaches either force subsets of the dictionary atoms to represent data from only specific classes [166], [222], [133] or they associate the complete dictionary to all the classes and constrain their sparse coefficients to be discriminative [234], [104], [137]. A third category of techniques learns exclusive sets of class specific and common dictionary atoms to separate the common and particular features of the data from different classes [203], [240]. Establishing association between the atoms and the corresponding class labels is a key step of the existing methods. However, adaptively building this association is still an open research problem [220]. Moreover, the strategy of assigning different number of dictionary atoms to different classes and adjusting the overall size of the
Chapter 10. Discriminative Bayesian Dictionary Learning for Classification

Figure 10.1: Schematics: For training, a set of probability distributions over the dictionary atoms, i.e. $\mathcal{N}$, is learned. We also infer sets of Bernoulli distributions indicating the probabilities of selection of the dictionary atoms in the expansion of data from each class. These distributions are used for inferring the support of the sparse codes. The (parameters of) Bernoulli distributions are later used for learning a classifier. The final dictionary is learned by sampling the distributions in $\mathcal{N}$, whereas the sparse codes are computed as element-wise product of the support and the weights (also inferred by the approach) of the codes. Combined, the dictionary and the codes faithfully represent the training data. For testing, sparse codes of the query over the dictionary are computed and fed to the classifier for labeling.

dictionary become critical for the classification accuracy of the existing approaches, as no principled approach is generally provided to predetermine these parameters.

In this work, we propose a solution to this problem by approaching the sparse representation based classification from a non-parametric Bayesian perspective. We propose a Bayesian sparse representation technique that infers a discriminative dictionary using a finite approximation of the Beta Process [157]. Our approach adaptively builds the association between the dictionary atoms and the class labels such that this association signifies the probability of selection of the dictionary atoms in the expansion of class-specific data. Furthermore, the non-parametric character of the approach allows it to automatically infer the correct size of the dictionary. The scheme employed by our approach is shown in Fig. 10.1. We perform Bayesian
inference over a model proposed for the discriminative sparse representation of the training data. The inference process learns distributions over the dictionary atoms and sets of Bernoulli distributions associating the dictionary atoms to the labels of the data. The Bernoulli distributions govern the support of the final sparse codes and are later utilized in learning a multi-class linear classifier. The final dictionary is learned by sampling the distributions over the dictionary atoms and the corresponding sparse codes are computed by an element-wise product of the support and the inferred weights of the codes. The learned dictionary and the sparse codes also represent the training data faithfully.

A query is classified in our approach by first sparsely encoding it over the inferred dictionary and then classifying its sparse code with the learned classifier. In this work, we learn the classifier and the dictionary using the same hierarchical Bayesian model. This allows us to exploit the aforementioned Bernoulli distributions in the accurate estimate of the classifier. We present the proposed Bayesian model along its inference equations for Gibbs sampling. Our approach has been tested on two face-databases [140], [77], an object-database [74], an action-database [172] and a scene-database [121]. The classification results are compared with the state-of-the-art discriminative sparse representation approaches. The proposed approach not only outperforms these approaches in terms of accuracy, its computational efficiency for the classification stage is also comparable to the most efficient existing approaches.

This paper is organized as follows. We review the related work in Section 10.2. In Section 10.3, we formulate the problem and explain the relevant concepts to clarify the rationale behind our approach. The proposed approach is presented in Section 10.4. Experimental results are reported in Section 10.5 and a discussion on the parameter settings is provided in Section 10.6. We draw conclusions in Section 10.7.

## 10.2 Related Work

There are three main categories of the approaches that learn discriminative sparse representation. In the first category, the learned dictionary atoms have a direct correspondence to the labels of the classes [222], [133], [166], [187], [204], [214], [38]. Yang et al. [222] proposed an SRC like framework for face recognition, where the atoms of the dictionary are learned from the training data instead of directly using the training data as the dictionary. In order to learn a dictionary that is simultaneously discriminative and reconstructive, Mairal et al. [133] used a discriminative penalty term in the K-SVD model [2], achieving state-of-the-art results on texture segmentation. Sprechmann and Sapiro [187] also proposed to learn dictionaries and
sparse codes for clustering. In [38], Castrodad and Sapiro computed class-specific dictionaries for actions. The dictionary atoms and their sparse coefficients also exploited the non-negativity of the signals in their approach. Active basis models are learned from the training images of each class and applied to object detection and recognition in [214]. Ramirez et al. [166] have used an incoherence promoting term for the dictionary atoms in their learning model. Encouraging incoherence among the class-specific sub-dictionaries allowed them to represent samples from the same class better than the samples from the other classes. Wang et al. [204] have proposed to learn class-specific dictionaries for modeling individual actions for action recognition. Their model incorporated a similarity constrained term and a dictionary incoherence term for classification. The above mentioned methods mainly associate a dictionary atom directly to a single class. Therefore, a query is generally assigned the label of the class whose associated atoms result in the minimum representational error for the query. The classification stages of the approaches under this category often require the computation of representations of the query over many sub-dictionaries.

In the second category, a single dictionary is shared by all the classes, however the representation coefficients are forced to be discriminative ([104], [137], [234], [218], [171], [159], [132], [125], [165], [105]). Jiang et al. [104] proposed a dictionary learning model that encourages the sparse representation coefficients of the same class to be similar. This is done by adding a 'discriminative sparse-code error' constraint to a unified objective function that already contains reconstruction error and classification error constraints. A similar approach is taken by Rodriguez and Sapiro [171] where the authors solve for a simultaneous sparse approximation problem [199] while learning the coefficients. It is common to learn dictionaries jointly with a classifier. Pham and Venkatesh [159] and Mairal et al. [137] proposed to train linear classifiers along the joint dictionaries learned for all the classes. Zhang and Li [234] enhanced the K-SVD algorithm [2] to learn a linear classifier along the dictionary. A task driven dictionary learning framework has also been proposed [132]. Under this framework, different risk functions of the representation coefficients are minimized for different tasks. Broadly speaking, the above mentioned approaches aim at learning a single dictionary together with a classifier. The query is classified by directly feeding its sparse codes over the learned single dictionary to the classifier. Thus, in comparison to the approaches in the first category, the classification stage of these approaches is computationally more efficient. In terms of learning a single dictionary for the complete training data and the classification stage, the proposed approach
also falls under this category of discriminative sparse representation techniques.

The third category takes a hybrid approach for learning the discriminative sparse representation. In these approaches, the dictionaries are designed to have a set of shared atoms in addition to class-specific atoms. Deng et al. [63] extended the SRC algorithm by appending an intra-class face variation dictionary to the training data. This extension achieves promising results in face recognition with a single training sample per class. Zhou and Fan [240] employ a Fisher-like regularizer on the representation coefficients while learning a hybrid dictionary. Wang and Kong [203] learned a hybrid dictionary to separate the common and particular features of the data. Their approach also encouraged the class-specific dictionaries to be incoherent. Shen et al. [181] proposed to learn a multi-level dictionary for hierarchical visual categorization. To some extent, it is possible to reduce the size of the dictionary using the hybrid approach, which also results in reducing the classification time in comparison to the approaches that fall under the first category. However, it is often non-trivial to decide on how to balance between the shared and the class-specific parts of the hybrid dictionary [221], [220].

The above mentioned approaches make the dictionaries discriminative by controlling the extent of their atom-sharing among class-specific representations. In this regard, latent variable models [61], [13], [127], [114] are also related to the discriminative dictionary learning framework. Damianou et al. [61] presented a Bayesian model that factorizes the latent variable space to represent shared and private information from multiple data views. They kept the segmentation of the latent space soft, such that a latent variable is even allowed to be more important to the shared space than the private space. Andrade-Pacheco et al. [13] later extended their approach to non-Gaussian data. Lu and Tang [127] also extended the Relevance Manifold Determination (RMD) [61] to learn face prior for Bayesian face recognition. Their approach first learned identity subspaces for each subject using RMD and later used the structure of the subspaces to estimate the Gaussian mixture densities in the observation space. Klami et al. [114] proposed a model for group factor analysis and formulated its solution as a variational inference of a latent variable model with structural sparsity.

### 10.3 Problem Formulation and Background

Let \( X = [X^1, ..., X^c, ..., X^C] \in \mathbb{R}^{m \times N} \) be the training data comprising \( N \) instances from \( C \) classes, wherein \( X^c \in \mathbb{R}^{m \times N_c} \) represents the data from the \( c^{th} \) class and \( \sum_{c=1}^{C} N_c = N \). The columns of \( X^c \) are indexed in \( I_c \). We denote a dictionary
by $\Phi \in \mathbb{R}^{m \times |K|}$ with atoms $\varphi_k$, where $k \in K = \{1, ..., K\}$ and $|$ represents the cardinality of the set. Let $A \in \mathbb{R}^{|K| \times N}$ be the sparse code matrix of the data, such that $X \approx \Phi A$. We can write $A = [A^1, ..., A^c, ..., A^C]$, where $A^c \in \mathbb{R}^{|K| \times |I^c|}$ is the sub-matrix related to the $c^{th}$ class. The $i^{th}$ column of $A$ is denoted as $\alpha_i \in \mathbb{R}^{|K|}$. To learn a sparse representation of the data, we can solve the following optimization problem:

$$<\Phi, A > = \min_{\Phi, A} ||X - \Phi A||_F^2 \quad s.t. \quad \forall i, ||\alpha_i||_p \leq t,$$

where $t$ is a predefined constant, $||.||_F$ computes the Frobenius norm and $||.||_p$ denotes the $\ell_p$-norm of a vector. Generally, $p$ is chosen to be 0 or 1 for sparsity [70]. The non-convex optimization problem of Eq. (10.1) can be iteratively solved by fixing one parameter and solving a convex optimization problem for the other parameter in each iteration. The solution to Eq. (10.1), factors the training data $X$ into two complementary matrices, namely the dictionary and the sparse codes, without considering the class label information of the training data. Nevertheless, we can still exploit this factorization in classification tasks by using the sparse codes of the data as features [104], for which, a classifier can be obtained as

$$W = \min_W \sum_{i=1}^N \mathcal{L}\{h_i, f(\alpha_i, W)\} + \lambda ||W||_F^2,$$

where $W \in \mathbb{R}^{C \times |K|}$ contains the model parameters of the classifier, $\mathcal{L}$ is the loss function, $h_i$ is the label of the $i^{th}$ training instance $x_i \in \mathbb{R}^m$ and $\lambda$ is the regularizer.

It is usually suboptimal to perform classification based on sparse codes learned by an unsupervised technique. Considering this, existing approaches [234], [159], [218], [137] proposed to jointly optimize a classifier with the dictionary while learning the sparse representation. One intended ramification of this approach is that the label information also gets induced into the dictionary. This happens when the information is utilized in computing the sparse codes of the data, which in turn are used for computing the dictionary atoms. This results in improving the discriminative abilities of the learned dictionary. Jiang et al. [104] built further on this concept and encouraged explicit correspondence between the dictionary atoms and the class-labels. More precisely, the following optimization problem is solved by the Label-Consistent K-SVD (LC-KSVD2) algorithm [104]:

$$<\Phi, W, T, A > = \min_{\Phi, W, T, A} \left\| \begin{pmatrix} X \\ \sqrt{\nu}Q \\ \sqrt{\nu}H \end{pmatrix} - \begin{pmatrix} \Phi \\ \sqrt{\nu}T \\ \sqrt{\nu}W \end{pmatrix} A \right\|_F^2 \quad s.t. \quad \forall i, ||\alpha_i||_0 \leq t$$

(10.3)
10.3. Problem Formulation and Background

(a) AR database [140]  
(b) Extended YaleB database [77]

Figure 10.2: Examples of how recognition accuracy is affected with varying dictionary size: $\kappa = 0$ for LC-KSVD1 and $\upsilon = 0$ for D-KSVD in Eq. (10.3). All other parameters are kept constant at optimal values reported in [104]. For the AR database, 2000 training instances are used and testing is performed with 600 instances. For the Extended YaleB, half of the database is used for training and the other half is used for testing. The instances are selected uniformly at random.

where $\upsilon$ and $\kappa$ are the regularization parameters, the binary matrix $\mathbf{H} \in \mathbb{R}^{C \times N}$ contains the class label information\(^1\), $\mathbf{T} \in \mathbb{R}^{|\mathcal{K}| \times |\mathcal{K}|}$ is the transformation between the sparse codes and the *discriminative sparse codes* $\mathbf{Q} \in \mathbb{R}^{|\mathcal{K}| \times N}$. Here, for the $i$\textsuperscript{th} training instance, the $i$\textsuperscript{th} column of the fixed binary matrix $\mathbf{Q}$ has 1 appearing at the $k$\textsuperscript{th} index only if the $k$\textsuperscript{th} dictionary atom has the same class label as the training instance. Thus, the discriminative sparse codes form a pre-defined relationship between the dictionary atoms and the class labels. This brings improvement to the discriminative abilities of the dictionary learned by solving Eq. (10.3).

It is worth noting that in Label-Consistent K-SVD algorithm [104], the relationship between class-specific subsets of dictionary atoms and class labels is pre-defined. However, regularization allows flexibility in this association during optimization. We also note that using $\upsilon = 0$ in Eq. (10.3) reduces the optimization problem to the one solved by Discriminative K-SVD (D-KSVD) algorithm [234]. Successful results are achievable using the above mentioned techniques for recognition and classification. However, like any discriminative sparse representation approach, these results are obtainable only after careful optimization of the algorithm parameters, including the dictionary size. In Fig. 10.2, we illustrate the behavior of recognition accuracy under varying dictionary sizes for [234] and [104] for two face databases.

Paisley and Carin [157] developed a *Beta Process* for non-parametric factor anal-

\(^1\)For the $i$\textsuperscript{th} training instance, the $i$\textsuperscript{th} column of $\mathbf{H}$ has 1 appearing only at the index corresponding to the class label.
ysis, which was later used by Zhou et al. [238] in successful image restoration. Exploiting the non-parametric Bayesian framework, a Beta Process can automatically infer the factor/dictionary size from the training data. With the base measure $h_0$ and parameters $a_0 > 0$ and $b_0 > 0$, a Beta Process is denoted as $BP(a_0, b_0, h_0)$. Paisley and Carin [157] noted that a finite representation of the process can be given as:

$$h = \sum_k \pi_k \delta_{\varphi_k}(\varphi), \quad k \in K = \{1, ..., K\},$$

$$\pi_k \sim \text{Beta}(\pi_k|a_0/K, b_0(K - 1)/K),$$

$$\varphi_k \sim h_0,$$  \hspace{1cm} (10.4)

In Eq. (10.4), $\delta_{\varphi_k}(\varphi)$ is 1 when $\varphi = \varphi_k$ and 0 otherwise. Therefore, a draw $h$ from the process can be represented as a set of $|K|$ probabilities, each having an associated vector $\varphi_k$, drawn i.i.d. from the base measure $h_0$. Using $h$, we can draw a binary vector $z_i \in \{0, 1\}^{|K|}$, such that the $k^{th}$ component of $z_i$ is drawn $z_{ik} \sim \text{Bernoulli}(\pi_k)$. By independently drawing $N$ such vectors, we may construct a matrix $Z \in \{0, 1\}^{|K| \times N}$, where $z_i$ is the $i^{th}$ column of this matrix.

Using the above mentioned finite approximation of the Beta Process, it is possible to factorize $X$ as follows:

$$X = \Phi Z + E,$$  \hspace{1cm} (10.5)

where, $\Phi \in \mathbb{R}^{m \times |K|}$ has $\varphi_k$ as its columns and $E \in \mathbb{R}^{m \times N}$ is the error matrix. In Eq. (10.5), the number of non-zero components in a column of $Z$ can be controlled by the parameters $a_0$ and $b_0$ in Eq. (10.4). The components of the $k^{th}$ row of $Z$ are independent draws from $\text{Bernoulli}(\pi_k)$. Let $\pi \in \mathbb{R}^{|K|}$ be a vector with $\pi_k \in \mathbb{R}$, as its $K^{th}$ element. This vector governs the probability of selection of the columns of $\Phi$ in the expansion of the data. Existence of this physically meaningful latent vector in the Beta Process based matrix factorization plays a central role in the proposed approach.

### 10.4 Proposed Approach

We propose a Discriminative Bayesian Dictionary Learning approach for classification. For the $c^{th}$ class, our approach draws $|I_c|$ binary vectors $z_i^c \in \mathbb{R}^{|K|}$, $\forall i \in I_c$ using a finite approximation of the Beta Process. For each class, the vectors are sampled using separate draws with the same base. That is, the matrix factorization is governed by a set of $C$ probability vectors $\pi^{c} \in \{1, ..., C\}$, instead of a single vector, however the inferred dictionary is shared by all the classes. An element of the
10.4. Proposed Approach

The aforementioned set, i.e. \( \pi^c \in \mathbb{R}^{|K|} \), controls the probability of selection of the dictionary atoms for a single class data. This promotes discrimination in the inferred dictionary.

10.4.1 The Model

Let \( \alpha^c_i \in \mathbb{R}^{|K|} \) denote the sparse code of the \( i^{th} \) training instance of the \( c^{th} \) class, i.e. \( x^c_i \in \mathbb{R}^m \), over a dictionary \( \Phi \in \mathbb{R}^{m \times |K|} \). Mathematically, \( x^c_i = \Phi \alpha^c_i + \epsilon_i \), where \( \epsilon_i \in \mathbb{R}^m \) denotes the modeling error. We can directly use the Beta Process discussed in Section 10.3 for computing the desired sparse code and the dictionary. However, the model employed by the Beta Process is restrictive, as it only allows the code to be binary. To overcome this restriction, let \( \alpha^c_i = z^c_i \odot s^c_i \), where \( \odot \) denotes the Hadamard/element-wise product, \( z^c_i \in \mathbb{R}^{|K|} \) is the binary vector and \( s^c_i \in \mathbb{R}^{|K|} \) is a weight vector. We place a standard normal prior \( N(s^c_{ik} | 0, 1/\lambda_{s_o}^c) \) on the \( k^{th} \) component of the weight vector \( s^c_{ik} \), where \( \lambda_{s_o}^c \) denotes the precision of the distribution. In here, as in the following text, we use the subscript ‘\( o \)’ to distinguish the parameters of the prior distributions. The prior distribution over the \( k^{th} \) component of the binary vector is \( \text{Bernoulli}(z^c_{ik} | \pi^c_k) \). We draw the atoms of the dictionary from a multivariate Gaussian base, i.e. \( \varphi_k \sim N(\varphi_k | \mu_{k_o}, \Lambda_{k_o}^{-1}) \), where \( \mu_{k_o} \in \mathbb{R}^m \) is the mean vector and \( \Lambda_{k_o} \in \mathbb{R}^{m \times m} \) is the precision matrix for the \( k^{th} \) atom of the dictionary. We model the error as zero mean Gaussian in \( \mathbb{R}^m \). Thus, we arrive at the following representation model:

\[
\begin{align*}
  x^c_i &= \Phi \alpha^c_i + \epsilon_i \quad \forall i \in I_c, \forall c \\
  \alpha^c_i &= z^c_i \odot s^c_i \\
  z^c_{ik} &\sim \text{Bernoulli}(z^c_{ik} | \pi^c_k) \\
  s^c_{ik} &\sim N(s^c_{ik} | 0, 1/\lambda_{s_o}^c) \\
  \pi^c_k &\sim \text{Beta}(\pi^c_k | a_o/K, b_o(K-1)/K) \\
  \varphi_k &\sim N(\varphi_k | \mu_{k_o}, \Lambda_{k_o}^{-1}) \quad \forall k \in K \\
  \epsilon_i &\sim N(\epsilon_i | 0, \Lambda_{\epsilon_o}^{-1}) \quad \forall i \in \{1, ..., N\} 
\end{align*}
\]

(10.6)

Notice, in the above model a conjugate Beta prior is placed over the parameter of the Bernoulli distribution, as mentioned in Section 10.3. Hence, a latent probability vector \( \pi^c \) (with \( \pi^c_k \) as its components) is associated with the dictionary atoms for the representation of the data from the \( c^{th} \) class. The common dictionary \( \Phi \) is inferred from \( C \) such vectors. In the above model, this fact is notationally expressed by showing the dictionary atoms being sampled from a common set of \( |K| \) distributions, while distinguishing the class-specific variables in the other notations with a
superscript ‘c’. We assume the same statistics for the modeling error over the complete training data\(^2\). We further place non-informative Gamma hyper-priors over the precision parameters of the normal distributions. That is, \(\lambda^c_s \sim \Gamma(\lambda^c_s|c_o, d_o)\) and \(\lambda^c \sim \Gamma(\lambda^c|e_o, f_o)\), where \(c_o, d_o, e_o\) and \(f_o\) are the parameters of the respective Gamma distributions. Here, we allow the error to have an isotropic precision, i.e. \(\Lambda^c = \lambda^c I_m\), where \(I_m\) denotes the identity matrix in \(\mathbb{R}^{m \times m}\). The graphical representation of the complete model is shown in Fig. 10.3.

10.4.2 Inference

Gibbs sampling is used to perform Bayesian inference over the proposed model\(^3\). Starting with the dictionary, below we derive analytical expressions for the posterior distributions over the model parameters for the Gibbs sampler. The inference process performs sampling over these posterior distributions. The expressions are derived assuming zero mean Gaussian prior over the dictionary atoms, with isotropic

---

\(^2\)It is also possible to use different statistics for different classes, however, in practice the assumption of similar noise statistics works well. We adopt the latter to avoid unnecessary complexity.

\(^3\)Paisley and Carin [157] derived variational Bayesian algorithm [15] for their model. It was shown by Zhou et al. [238] that Gibbs sampling is an equally effective strategy in data representation using the same model. Since it is easier to relate the Gibbs sampling process to the learning process of conventional optimization based sparse representation (e.g. K-SVD [2]), we derive expressions for the Gibbs sampler for our approach. Due to the conjugacy of the model, these expressions can be derived analytically.
10.4. Proposed Approach

precision. That is, \( \mu_{ko} = 0 \) and \( \Lambda_{ko} = \lambda_{ko} I_m \). This simplification leads to faster sampling, without significantly affecting the accuracy of the approach. The sampling process samples the atoms of the dictionary one-by-one from their respective posterior distributions. This process is analogous to the atom-by-atom dictionary update step of K-SVD [2], however the sparse codes remain fixed during our dictionary update.

**Sampling \( \varphi_k \):** For our model, we can write the following about the posterior distribution over a dictionary atom:

\[
p(\varphi_k | -) \propto \prod_{i=1}^{N} N(x_i | \Phi(z_i \odot s_i), \lambda_{eo}^{-1} I_m) N(\varphi_k | 0, \lambda_{ko}^{-1} I_m).
\]

Here, we intentionally dropped the superscript ‘c’ as the dictionary is updated using the complete training data. Let \( x_{i\varphi_k} \) denote the contribution of the dictionary atom \( \varphi_k \) to the \( i \)th training instance \( x_i \):

\[
x_{i\varphi_k} = x_i - \Phi(z_i \odot s_i) + \varphi_k (z_{ik} \odot s_{ik}). \tag{10.7}
\]

Using Eq. (10.7), we can re-write the aforementioned proportionality as

\[
p(\varphi_k | -) \propto \prod_{i=1}^{N} N(x_{i\varphi_k} | \varphi_k (z_{ik} s_{ik}), \lambda_{eo}^{-1} I_m) N(\varphi_k | 0, \lambda_{ko}^{-1} I_m).
\]

Considering the above expression, the posterior distribution over a dictionary atom can be written as

\[
p(\varphi_k | -) = N(\varphi_k | \mu_k, \lambda_k^{-1} I_m), \tag{10.8}
\]

where,

\[
\mu_k = \frac{\lambda_{eo}}{\lambda_k} \sum_{i=1}^{N} (z_{ik}.s_{ik}) x_{i\varphi_k}, \quad \lambda_k = \lambda_{ko} + \lambda_{eo} \sum_{i=1}^{N} (z_{ik}.s_{ik})^2.
\]

**Sampling \( z_{ik}^c \):** Once the dictionary atoms have been sampled, we sample \( z_{ik}^c \), \( \forall i \in \mathcal{I}_c, \forall k \in \mathcal{K} \). Using the contribution of the \( k \)th dictionary atom, the posterior probability distribution over \( z_{ik}^c \) can be expressed as

\[
p(z_{ik}^c | -) \propto N(x_{i\varphi_k}^c | \varphi_k (z_{ik}^c s_{ik}^c), \lambda_{eo}^{-1} I_m) \text{Bernoulli}(z_{ik}^c | \pi_{ko}^c).
\]

Here we are concerned with the \( c \)th class only, therefore \( x_{i\varphi_k}^c \) is computed with the \( c \)th class data in Eq. (10.7). With the prior probability of \( z_{ik}^c = 1 \) given by \( \pi_{ko}^c \), we can write the following about its posterior probability:

\[
p(z_{ik}^c = 1 | -) \propto \pi_{ko}^c \exp \left( -\frac{\lambda_{eo}}{2} ||x_{i\varphi_k}^c - \varphi_k s_{ik}^c||^2_2 \right).
\]
It can be shown that the right hand side of the proportionality can be written as:

\[ p_1 = \pi_{c_k} \zeta_1 \zeta_2, \]

where, \( \zeta_1 = \exp\left( -\frac{\lambda_o s_{ik}^2}{2} (||\varphi_k||^2 + 2(x_{ik}^c)^T \varphi_k) \right) \) and \( \zeta_2 = \exp\left( -\frac{\lambda_o}{2} ||x_{ik}^c||^2 \right) \).

Furthermore, since the prior probability of \( z_{ik}^c = 0 \) is given by \( 1 - \pi_{c_k} \), we can write the following about its posterior probability:

\[ p(z_{ik}^c = 0 | -) \propto (1 - \pi_{c_k}) \zeta_2. \]

Thus, \( z_{ik}^c \) can be sampled from the following normalized Bernoulli distribution:

\[ \text{Bernoulli}\left( z_{ik}^c | \frac{\pi_{c_k} \zeta_1}{1 + \pi_{c_k} (\zeta_1 - 1)} \right). \] (10.9)

**Sampling \( s_{ik}^c \):** We can write the following about the posterior distribution over \( s_{ik}^c \):

\[ p(s_{ik}^c | -) \propto \mathcal{N}(x_{ik}^c | \varphi_k(z_{ik}^c s_{ik}^c), \lambda_o^{-1} I_m)\mathcal{N}(s_{ik}^c | 0, 1/\lambda_o). \]

Again, notice that we are concerned with the \( c \)-th class data only. In light of the above expression, \( s_{ik}^c \) can be sampled from the following posterior distribution:

\[ p(s_{ik}^c | -) = \mathcal{N}(s_{ik}^c | \mu_c^c, 1/\lambda_c^s), \] (10.10)

where, \( \mu_c^c = \lambda_o \lambda_c^c z_{ik}^c \varphi_k x_{ik}^c \), \( \lambda_c^s = \lambda_o \lambda_c^c + \lambda_o (z_{ik}^c)^2 ||\varphi_k||^2 \).

**Sampling \( \pi_k^c \):** Based on our model, we can also write the posterior probability distribution over \( \pi_k^c \) as

\[ p(\pi_k^c | -) \propto \prod_{i \in I_c} \text{Bernoulli}(z_{ik}^c | \pi_{c_k})\text{Beta}\left( \pi_{c_k} | \frac{a_o}{K}, \frac{b_o(K - 1)}{K} \right). \]

Using the conjugacy between the distributions, it can be easily shown that the \( k \)-th component of \( \pi^c \) must be drawn from the following posterior distribution during the sampling process:

\[ p(\pi_k^c | -) = \text{Beta}\left( \frac{a_o}{K} + \sum_{i \in I_c} z_{ik}^c, \frac{b_o(K - 1)}{K} + |I_c| - \sum_{i \in I_c} z_{ik}^c \right). \] (10.11)
10.4. Proposed Approach

**Sampling** $\lambda^c$: In our model, the components of the weight vectors are drawn from a standard normal distribution. For a given weight vector, common priors are assumed over the precision parameters of these distributions. This allows us to express the likelihood function for $\lambda^c$ in terms of standard multivariate Gaussian with isotropic precision. Thus, we can write the posterior over $\lambda^c$ as the following:

$$p(\lambda^c|\cdot) \propto \prod_{i \in I_c} \mathcal{N}\left(s^c_i \mid 0, \frac{1}{\lambda^c_{s_o} I_{|K|}} \right) \Gamma(\lambda^c_{s_o}|c_o, d_o).$$

Using the conjugacy between the Gaussian and Gamma distributions, it can be shown that $\lambda^c$ must be sampled as:

$$\lambda^c \sim \Gamma\left(\lambda^c_{s} | \frac{|K|N_c}{2} + c_o, \frac{1}{2} \sum_{i \in I_c} ||s^c_i||^2_2 + d_o\right). \quad (10.12)$$

**Sampling** $\lambda_e$: We can write the posterior over $\lambda_e$ as

$$p(\lambda_e|\cdot) \propto \prod_{i=1}^{N} \mathcal{N}(x_i|\Phi(z_i \odot s_i), \lambda^{-1}_e I_m) \Gamma(\lambda_e|c_o, f_o).$$

Similar to $\lambda^c$, we can arrive at the following for sampling $\lambda_e$ during the inferencing process:

$$\lambda_e \sim \Gamma\left(\frac{mN}{2} + c_o, \frac{1}{2} \sum_{i=1}^{N} ||x_i - \Phi(z_i \odot s_i)||^2_2 + f_o\right). \quad (10.13)$$

As a result of Bayesian inference, we obtain sets of posterior distributions over the model parameters. We are particularly interested in two of them. Namely, the set of distributions over the dictionary atoms $\mathcal{N} = \{\mathcal{N}(\varphi_k|\mu_k, \Lambda^{-1}_k) : k \in K\} \subset \mathbb{R}^m$, and the set of probability distributions characterized by the vectors $\pi_{c \in \{1,...,C\}} \in \mathbb{R}^{|K|}$. Momentarily, we defer the discussion on the latter. The former is used to compute the desired dictionary $\Phi$. This is done by drawing multiple samples from the elements of $\mathcal{N}$ and estimating the corresponding dictionary atoms as respective means of the samples. Indeed, the mean parameters of the elements of $\mathcal{N}$ can also be chosen as the desired dictionary atoms. However, we adopt the former approach because it also accounts for the precisions of the posterior distributions while computing the final dictionary. Although the difference in the performance resulting from the two approaches is generally very small, the adopted approach is preferred as it adds to the robustness of the dictionary against the noise in the training data [5].
Our model allows to estimate the desired size of the dictionary non-parametrically. We present Lemma 5 regarding the expected size of the dictionary according to our model. In Lemma 6, we make an observation that is exploited in the sampling process to estimate this size.

**Lemma 5.** For a very large $K$, $\mathbb{E}[\xi] = \frac{a_0}{b_0}$, where $\xi = \sum_{k=1}^{K} z_{ik}^c$.

**Proof:** According to the proposed model, the covariance of a data vector from the $c$th class, i.e. $x_i^c$ can be given by:

$$\mathbb{E}[(x_i^c)(x_i^c)^T] = \frac{a_o K}{a_o + b_o(K-1)} \frac{\Lambda^{-1}_{ko}}{\lambda^o_k} + \Lambda^{-1}_{c_o}. \tag{10.14}$$

In Eq. (10.14), fraction $\frac{a_o}{b_o}$ appears due to the presence of $z_{ik}^c$ in the model and the equation simplifies to $\mathbb{E}[(x_i^c)(x_i^c)^T] = \frac{a_o K}{b_o(K-1)} \frac{\Lambda^{-1}_{ko}}{\lambda^o_k} + \Lambda^{-1}_{c_o}$ when we neglect $z_{ik}^c$. Here, $K$ signifies the number of dictionary atoms required to represent the data vector. In the equation, as $K$ becomes very large, $\mathbb{E}[(x_i^c)(x_i^c)^T] \to \frac{a_o}{b_o} \frac{\Lambda^{-1}_{ko}}{\lambda^o_k} + \Lambda^{-1}_{c_o}$. Thus, for a large dictionary, the expected number of atoms required to represent $x_i^c$ is given by $\frac{a_o}{b_o}$. Meaning, $\mathbb{E}[\xi] = \frac{a_o}{b_o}$, where $\xi = \sum_{k=1}^{K} z_{ik}^c$.

**Lemma 6.** Once $\pi_k^c = 0$ in a given iteration of the sampling process, $\mathbb{E}[\pi_k^c] \approx 0$ for the later iterations.

**Proof:** According to Eq. (10.9), $\forall i \in \mathcal{I}_c$, $z_{ik}^c = 0$ when $\pi_k^c = 0$. Once this happens, the posterior distribution over $\pi_k^c$ becomes Beta $\left( \frac{\pi_k^c}{\bar{a}, \bar{b}} \right)$, where $\bar{a} = \frac{a_o}{K}$ and $\bar{b} = \frac{b_o(K-1)}{K} + |\mathcal{I}_c|$ (see Eq. 10.11). Thus, the expected value of $\pi_k^c$ for the later iterations can be written as $\mathbb{E}[\pi_k^c] = \frac{\bar{a}}{\bar{a} + \bar{b}} = \frac{a_o}{a_o + b_o(K-1) + K|\mathcal{I}_c|}$. With $0 < a_o, b_o < |\mathcal{I}_c| \ll K$ we can see that $\mathbb{E}[\pi_k^c] \approx 0$.

Considering Lemma 5, we start with a very large value of $K$ in the Gibbs sampling process. We let $K = 1.5 \times N$ and let $0 < a_o, b_o < |\mathcal{I}_c|$ to ensure that the resulting representation is sparse. We drop the $k$th dictionary atom during the sampling process if $\pi_k^c = 0$, for all the classes simultaneously. According to Lemma 6, dropping such an atom does not bring significant changes to the final representation. Thus, by removing the redundant dictionary atoms in each sampling iteration, we finally arrive at the correct size of the dictionary, i.e. $|\mathcal{K}|$.

As mentioned above, with Bayesian inference over the proposed model we also infer a set of probability vectors $\pi^{c \in \{1, \ldots, C\}}$. Each element of this set, i.e. $\pi^c \in \mathbb{R}^{|\mathcal{K}|}$, we follow [157] closely in the proof, however, our analysis also takes into account the class labels of the data, whereas no such data discrimination is assumed in [157].
further characterizes a set of probability distributions $\mathcal{G}^c = \{\text{Bernoulli}(\pi_k^c) \colon k \in K\} \subset \mathbb{R}$. Here, Bernoulli($\pi_k^c$) is jointly followed by all the $k^{th}$ components of the sparse codes for the $c^{th}$ class. If the $k^{th}$ dictionary atom is commonly used in representing the $c^{th}$ class training data, we must expect a high value of $\pi_k^c$, and $\pi_k^c \rightarrow 0$ otherwise. In other words, for an arranged dictionary, components of $\pi^c$ having large values should generally cluster well if the learned dictionary is discriminative. Furthermore, these clusters must appear at different locations in the inferred vectors for different classes. Such clusterings would demonstrate the discriminative character of the inferred dictionary. Figure 10.4 verifies this character for the dictionaries inferred under the proposed model. Each row of the figure plots six different probability vectors (i.e. $\pi^c$) for different training datasets. A clear clustering of the high value components of the vectors is visible in each plot. In Appendix D.1, we also illustrate few dictionary atoms corresponding to the largest values of $\pi_k^c$ for the Extended YaleB database [77]. Detailed experiments are presented in Section 10.5.

Whereas clear clusters are visible in Fig. 10.4, we can also see few non-zero values appearing far from the main clusters. These values indicate the sharing of atoms among the data representations of different classes. We note that our model allows such sharing because it employs finite approximation of the Beta Process. Such a model is sufficient for practical classification tasks where the training data size is always finite and known a priori. Our model only requires $K$ to be larger than the training data size. We also note that the model does not allow the atom sharing between the classes if $K$ is infinite. In that case, an atom of the dictionary will correspond to only a single class. This is similar to the first category of the discriminative dictionary learning approaches discussed in Section 10.2.

### 10.4.3 Classification

To classify a query $y \in \mathbb{R}^m$, we first compute its sparse representation $\hat{\alpha}$ over the learned dictionary. The label of the query is predicted by maximizing the coefficients of the vector $\ell = W\hat{\alpha}$, where $W \in \mathbb{R}^{C \times |K|}$ denotes a multi-class linear classifier. Effectiveness of learning such a classifier in accordance with the dictionary is already established for discriminative dictionary learning [234], [104]. Therefore, we also couple our classifier with the learned dictionary. Nevertheless, we keep the learning processes of the dictionary and the classifier disjoint to fully exploit the potential of our model. Further discussion in this regard is deferred to Section 10.6. In order to learn the classifier, we first define a vector $h^c \in \mathbb{R}^C$ for each class. These vectors are computed using $\pi^c \in \{1, \ldots, C\}$ inferred by the dictionary learning process. For the
Figure 10.4: Illustration of the discriminative character of the inferred dictionary: From top, the four rows present results on AR database [140], Extended YaleB [77], Caltech-101[74] and Fifteen Scene categories [121], respectively. In each plot, the x-axis represents $k \in K$ and the y-axis shows the corresponding probability of selection of the $k^{th}$ dictionary atom in the expansion of the data. A plot represents a single $\pi^c$ vector learned as a result of Bayesian inference. For the first three rows, from left to right, the value of $c$ (i.e. class label) is 1, 5, 10, 20 and 25, respectively. For the fourth row the value of $c$ is 1, 3, 5, 7, 9 and 11 for the plots from left to right. Plots clearly show distinct clusters of high probabilities for different classes.

For the $c^{th}$ class, the $q^{th}$ coefficient $h_{iq}^c$ of $h^c$ is computed as $h_{iq}^c = \sum_{k \in C} \pi_k^q$ where, $C$ indexes the non-zero coefficients of $\pi^c$. Considering that large non-zero coefficients of $\pi^c$ generally appear at the locations corresponding to the $c^{th}$ class, $h_{iq}^c$ is large when $q = c$ and small otherwise. After normalization, we use the computed vectors as the training data for the classifier. The training is done by solving $\mathbf{h}_i^c = \mathbf{W}\beta_i^c + \epsilon_i$ using the model in Eq. 10.6. Here, $\beta_i^c \in \mathbb{R}^{|K|}$ is a sparse coefficient vector defined over $\mathbf{W}$, just as $\alpha_i^c$ was defined over the dictionary.

The inference process for learning the classifier is also guided by the probability vectors $\pi^{c \in \{1, \ldots, C\}}$ computed by the dictionary learning stage. We directly use these vectors for classifier learning and keep them fixed during the complete sampling process. Notice that, our sampling process computes a basis keeping in view the support of its coefficient vectors, i.e. $\varphi_k$ depends on $z_{ik}$ in Eq. 10.8. Since the support of $\alpha_i^c$ and $\beta_i^c$ follow the same set of probability distributions, given by $\pi^{c \in \{1, \ldots, C\}}$, a coupling is induced between their inferred bases, i.e. $\Phi$ and $\mathbf{W}$. This forces the learned parameters of $\mathbf{W}$ to respect the popularity of the dictionary atoms for
representing the class-specific training data. Since the popularity of the atoms is expected to remain consistent across the training and the test data for a given class, we can directly use the classifier with the sparse codes of the test data to correctly predict its class label.

To classify a query, we first find its sparse representation over the learned dictionary. Keeping in view that our dictionary learning model imposes sparsity on a coefficient vector by forcing many of its components to zero, we choose Orthogonal Matching Pursuit (OMP) algorithm [33] to efficiently compute the sparse representation of the query signal. OMP allows only a few non-zero components in the representation vector to maximally approximate the query signal using the dictionary. Therefore, the popular dictionary atoms for the correct class of the query usually contribute significantly to the representation. This helps in accurate classification using $W$. Notice that, to predict the label of a $K$-dimensional sparse vector, our approach only has to multiply it with a $C \times K$-dimensional matrix and search for the maximum value in the resulting $C$-dimensional vector. This makes our classification approach much efficient compared to the alternative of using a sophisticated classifier like SVM to classify the sparse codes of the query. Since efficient classification of a query signal is one of the major goals of discriminative dictionary learning, we consider our approach highly attractive.

10.4.4 Initialization

For inferring the dictionary, we need to first initialize $\Phi$, $z^c_i$, $s^c_i$ and $\pi^c_k$. We initialize $\Phi$ by randomly selecting the training instances with replacement. We sparsely code $x^c_i$ over the initial dictionary using OMP [33]. The codes are considered as the initial $s^c_i$, whereas their support forms the initial vector $z^c_i$. Computing the initial $s^c_i$ and $z^c_i$ with other methods, such as regularized least squares, is equally effective. We set $\pi^c_k = 0.5, \forall c, \forall k$ for the initialization. Notice, this means that all the dictionary atoms initially have equal chances of getting selected in the expansion of a training instance from any class. The values of $\pi^c_k, \forall c, \forall k$ finally inferred by the dictionary learning process serve as the initial values of these parameters for learning the classifier. Similarly, $z^c_i$ and $s^c_i$ computed by the dictionary learning stage are used for initializing the corresponding vectors for the classifier. We initialize $W$ using the ridge regression [79] with the $\ell_2$-norm regularizer and quadratic loss:

$$W = \min_W ||H - W\alpha_i||^2 + \lambda||W||^2_2, \forall i \in \{1, ..., N\}, \quad (10.15)$$

where $\lambda$ is the regularization constant. The computation is done over the complete training data, therefore the superscript ‘$c$’ is dropped in the above equation. Similar
to the existing approaches [234], [104], we consider the initialization procedure as an integral part of the proposed approach.

10.5 Experiments

We have evaluated the proposed approach on two face data sets: the Extended YaleB [77] and the AR database [140], a data set for object categories: Caltech-101 [74], a data set for scene categorization: Fifteen scene categories [121], and an action data set: UCF sports actions [172]. These data sets are commonly used in the literature for evaluation of sparse representation based classification techniques. We compare the performance of the proposed approach with SRC [213], the two variants of Label-Consistent K-SVD [104] (i.e. LC-KSVD1, LC-KSVD2), the Discriminative K-SVD algorithm (D-KSVD) [234], the Fisher Discrimination Dictionary Learning algorithm (FDDL) [221] and the Dictionary Learning based on separating the Commonalities and the Particularities of the data (DL-COPAR) [203]. In our comparisons, we also include results of unsupervised sparse representation based classification that uses K-SVD [2] as the dictionary learning technique and separately computes a multi-class linear classifier using Eq. (10.15).

For all of the above mentioned methods, except SRC and D-KSVD, we acquired the public codes from the original authors. To implement SRC, we used the LASSO [194] solver of the SPAMS toolbox [135]. For D-KSVD, we used the public code provided by Jiang et al. [104] for LC-KSVD2 algorithm and solved Eq. 10.3 with $\nu = 0$. In all experiments, our approach uses the implementation of OMP made public by Elad et al. [177]. K-SVD, D-KSVD, LC-KSVD1 and LC-KSVD2 also use the same implementation. The experiments are performed on an Intel Core i7-2600 CPU at 3.4 GHz with 8 GB RAM. We performed our own experiments using the above mentioned methods and the proposed approach using the same data. The parameters of the existing approaches were carefully optimized following the guidelines of the original works. We mention the used parameter values and, where it exists, we note the difference between our values and those used in the original works. In our experiments, these differences were made to favor the existing approaches. Results of the approaches other than those mentioned above, are taken directly from the literature, where the same experimental protocol has been followed.

For the proposed approach, the used parameter values were as follows. In all experiments, we chose $K = 1.5N$ for initialization, whereas $c_o, d_o, e_o$ and $f_o$ were all set to $10^{-6}$. We selected $a_o = b_o = \min_{i \in I_o} |I_o|$, whereas $\lambda_{k_o}$ and $\lambda_{k_o}$ were set to 1 and $m$, respectively. Furthermore, $\lambda_{c_o}$ was set to $10^6$ for all the datasets except for
10.5. Experiments

Fifteen Scene Categories [121], where we used $\lambda_o = 10^9$. In each experiment, we ran 500 Gibbs sampling iterations that proved sufficient for accurate inference using our approach. We provide assessment of the inference accuracy of the performed Gibbs sampling in Appendix D.2. We defer further discussion on the selection of the parameter values to Section 10.6.

10.5.1 Extended YaleB

Extended YaleB [77] contains 2,414 frontal face images of 38 different people, each having about 64 samples. The images are acquired under varying illumination conditions and the subjects have different facial expressions. This makes the database fairly challenging, see Fig 10.5a for a few examples. In our experiments, we used the random face feature descriptor [213], where a cropped $192 \times 168$ pixels image was projected onto a 504-dimensional vector. For this, the projection matrix was generated from random samples of standard normal distributions. Following the common settings for this database, we chose one half of the images for training and the remaining samples were used for testing. We performed ten experiments by randomly selecting the samples for training and testing. Based on these experiments, the mean recognition accuracies of different approaches are reported in Table 10.1. The results for Locality-constrained Linear Coding (LLC) [206] is directly taken from [104], where the accuracy is computed using 70 local bases. Similar to Jiang et al. [104], the sparsity threshold for K-SVD, LC-KSVD1, LC-KSVD2 and D-KSVD was set to 30 in our experiments. Larger values of this parameter were found to be ineffective as they mainly resulted in slowing the algorithms without improving the recognition accuracy. Furthermore, as in [104], we used $\nu = 4.0$ for LC-KSVD1 and LC-KSVD2, whereas $\kappa$ was set to 2.0 for LC-KSVD2 and D-KSVD in Eq. (10.3). Keeping these parameter values fixed, we optimized for the number of dictionary atoms for each algorithm. This resulted in selecting 600 atoms for LC-KSVD2, D-KSVD and K-SVD, whereas 500 atoms consistently resulted in the best performance of LC-KSVD1. This value is set to 570 in [104] for all of the four methods. In all techniques that learn dictionaries, we used the complete training data in the learning process. Therefore, all training samples were used as dictionary atoms for SRC. Following [213], we set the residual error tolerance to 0.05 for SRC. Smaller values of this parameter also resulted in very similar accuracies. For FDDL, we followed [221] for the optimized parameter settings. These settings are the same as those reported for AR database in the original work. We refer the reader to the original work for the list of the parameters and their exact
Table 10.1: Recognition accuracy with Random-Face features on the Extended YaleB database [77]. The computed average time is for classification of a single instance.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy %</th>
<th>Average Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLC [206]</td>
<td>90.7</td>
<td>-</td>
</tr>
<tr>
<td>K-SVD [2]</td>
<td>93.13 ± 0.43</td>
<td>0.37</td>
</tr>
<tr>
<td>LC-KSVD1[104]</td>
<td>93.59 ± 0.54</td>
<td>0.36</td>
</tr>
<tr>
<td>D-KSVD [234]</td>
<td>94.79 ± 0.49</td>
<td>0.38</td>
</tr>
<tr>
<td>DL-COPAR [203]</td>
<td>94.83 ± 0.52</td>
<td>32.55</td>
</tr>
<tr>
<td>LC-KSVD2[104]</td>
<td>95.22 ± 0.61</td>
<td>0.39</td>
</tr>
<tr>
<td>FDDL [221]</td>
<td>96.07 ± 0.64</td>
<td>49.59</td>
</tr>
<tr>
<td>DBDL+SVM</td>
<td>96.10 ± 0.25</td>
<td>426.14</td>
</tr>
<tr>
<td>SRC [213]</td>
<td>96.32 ± 0.85</td>
<td>53.12</td>
</tr>
<tr>
<td>Proposed</td>
<td>97.31 ± 0.67</td>
<td>1.22</td>
</tr>
</tbody>
</table>

The results reported in the table are obtained by the Global Classifier (GC) of FDDL, which showed better performance than the Local Classifier (LC). For the parameter settings of DL-COPAR we followed the original work [203]. We fixed 15 atoms for each class and a set of 5 atoms was chosen to learn commonalities of the classes. The reported results are achieved by LC, that performed better than GC in our experiments.

It is clear from Table 10.1 that our approach outperforms the above mentioned approaches in terms of recognition accuracy, with nearly 23% improvement over the error rate of the second best approach. Furthermore, the time required by the proposed approach for classifying a single test instance is also very low as compared to SRC, FDDL and DL-COPAR. For the proposed approach, this time is comparable to D-KSVD and LC-KSVD. Like these algorithms, the computational efficiency in
the classification stage of our approach comes from using the learned multi-class linear classifier to classify the sparse codes of a test instance. To show the computational benefits of the proposed classifier over SVM, we also include the results of using SVM on the sparse code features of the query. In the table, DBDL+SVM refers to these results. Note that, our classifier also used the same features.

10.5.2 AR Database

This database contains more than 4,000 face images of 126 people. There are 26 images per person taken during two different sessions. The images in AR database have large variations in terms of facial expressions, disguise and illumination conditions. Samples from AR database are shown in Fig. 10.5b for illustration. We followed a common evaluation protocol in our experiments for this database, in which we used a subset of 2600 images pertaining to 50 males and 50 female subjects. For each subject, we randomly chose 20 samples for training and the rest for testing. The $165 \times 120$ pixel images were projected onto a 540-dimensional vector with the help of a random projection matrix, as in Section 10.5.1. We report the average recognition accuracy of our experiments in Table 10.2, which also includes the accuracy of LLC [206] reported in [104]. The mean values reported in the table are based on ten experiments.

In our experiments, we set the sparsity threshold for K-SVD, LC-KSVD1, LC-KSVD2 and D-KSVD to 50 as compared to 10 and 30 which was used in [234] and [104], respectively. Furthermore, the dictionary size for K-SVD, LC-KSVD2 and D-KSVD was set to 1500 atoms, whereas the dictionary size for LC-KSVD1 was set to 750. These large values (compared to 500 used in [234], [104]) resulted in better accuracies at the expense of more computation. However, the classification time per test instance remained reasonably small. In Table 10.2, we also include the results of LC-KSVD1, LC-KSVD2 and D-KSVD using the parameter values proposed in the original works. These results are distinguished with the $‡$ sign. For FDDL and DL-COPAR we used the same parameter settings as in Section 10.5.1. The reported results are for GC and LC for FDDL and DL-COPAR, respectively. For SRC we set the residual error tolerance to $10^{-6}$. This small value gave the best results.

From Table 10.2, we can see that the proposed approach performs better than the existing approaches in terms of accuracy. The recognition accuracies of SRC and FDDL are fairly close to our approach however, these algorithms require large amount of time for classification. This fact compromises their practicality. In contrast, the proposed approach shows high recognition accuracy (i.e. 22% reduction
Table 10.2: Recognition accuracy on the AR database [140]. The computed time is for classifying a single instance. The ‡ sign denotes the results using the parameter settings reported in the original works.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy %</th>
<th>Average Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLC [206]</td>
<td>88.7</td>
<td>-</td>
</tr>
<tr>
<td>DL-COPAR [203]</td>
<td>93.23 ± 1.71</td>
<td>39.80</td>
</tr>
<tr>
<td>LC-KSVD1[104]</td>
<td>93.48 ± 1.13</td>
<td>0.98</td>
</tr>
<tr>
<td>LC-KSVD1‡</td>
<td>87.48 ± 1.19</td>
<td>0.37</td>
</tr>
<tr>
<td>K-SVD [2]</td>
<td>94.13 ± 1.20</td>
<td>0.99</td>
</tr>
<tr>
<td>LC-KSVD2 [104]</td>
<td>95.33 ± 1.24</td>
<td>1.01</td>
</tr>
<tr>
<td>LC-KSVD2‡</td>
<td>88.35 ± 1.33</td>
<td>0.41</td>
</tr>
<tr>
<td>D-KSVD [234]</td>
<td>95.47 ± 1.50</td>
<td>1.01</td>
</tr>
<tr>
<td>D-KSVD‡</td>
<td>88.29 ± 1.38</td>
<td>0.38</td>
</tr>
<tr>
<td>DBDL+SVM</td>
<td>95.69 ± 0.73</td>
<td>104.01</td>
</tr>
<tr>
<td>FDDL [221]</td>
<td>96.22 ± 1.03</td>
<td>50.03</td>
</tr>
<tr>
<td>SRC [213]</td>
<td>96.65 ± 1.37</td>
<td>62.86</td>
</tr>
<tr>
<td>Proposed</td>
<td>97.47 ± 0.99</td>
<td>1.28</td>
</tr>
</tbody>
</table>

in the error rate as compared to SRC) with less than 1.5 ms required for classifying a test instance. The relative difference between the classification time of the proposed approach and the existing approaches remains similar in the experiments below. Therefore, we do not explicitly note these timings for all of the approaches in these experiments.

10.5.3 Caltech-101

The Caltech-101 database [74] comprises 9,144 samples from 102 classes. Among these, there are 101 object classes (e.g. minarets, trees, signs) and one “background” class. The number of samples per class varies from 31 to 800, and the images within a given class have significant shape variations, as can be seen in Fig. 10.6. To use the database, first the SIFT descriptors [126] were extracted from 16 × 16 image patches, which were densely sampled with a 6-pixels step size for the grid. Then, based on the extracted features, spatial pyramid features [121] were extracted with 2<sup>l</sup> × 2<sup>l</sup> grids, where l = 0, 1, 2. The codebook for the spatial pyramid was trained using k-means with k = 1024. Then, the dimension of a spatial pyramid feature was reduced to 3000 using PCA. Following the common experimental protocol, we selected 5, 10,
10.5. Experiments

Figure 10.6: Examples from Caltech-101 database [74]. The proposed approach achieves 100% accuracy on these classes.

15, 20, 25 and 30 instances for training the dictionary and the remaining instances were used in testing, in our six different experiments. Each experiment was repeated ten times with random selection of train and test data. The mean accuracies of these experiments are reported in Table 10.3.

For this dataset, we set the number of dictionary atoms used by K-SVD, LC-KSVD1, LC-KSVD2 and D-KSVD to the number of training examples available. This resulted in the best performance of these algorithms. The sparsity level was also set to 50 and $\nu$ and $\kappa$ were set to 0.001. Jiang et al. [104] also suggested the same parameter settings. For SRC, the error tolerance of $10^{-6}$ gave the best results in our experiments. We used the parameter settings for object categorization given in [221] for FDDL. For DL-COPAR, the selected number of class-specific atoms were kept the same as the number of training instances per class, whereas the number of shared atoms were fixed to 314, as in the original work [203]. For this database GC performed better than LC for DL-COPAR in our experiments.

From Table 10.3, it is clear that the proposed approach consistently outperforms the competing approaches. For some cases the accuracy of LC-KSVD2 is very close to the proposed approach, however with the increasing number of training instances the difference between the results increases in favor of the proposed approach. This is an expected phenomenon since more training samples result in more precise posterior distributions in Bayesian settings. We provide further discussion on dependence of our approach’s performance on training data size in Appendix D.3. Here, it is also worth mentioning that being Bayesian, the proposed approach is inherently an online technique. This means, in our approach, the computed posterior distributions can be used as prior distributions for further inference if more training data is available. Moreover, our approach is able to handle a batch of large training data
Table 10.3: Classification results on the Caltech-101 dataset [74].

<table>
<thead>
<tr>
<th>Total training samples</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhang et al. [232]</td>
<td>46.6</td>
<td>55.8</td>
<td>59.1</td>
<td>62.0</td>
<td>-</td>
<td>66.20</td>
</tr>
<tr>
<td>Lazebnik et al. [121]</td>
<td>-</td>
<td>-</td>
<td>56.4</td>
<td>-</td>
<td>-</td>
<td>64.6</td>
</tr>
<tr>
<td>Griffin et al. [82]</td>
<td>44.2</td>
<td>54.5</td>
<td>59.0</td>
<td>63.3</td>
<td>65.8</td>
<td>67.6</td>
</tr>
<tr>
<td>Wang et al. [206]</td>
<td>51.1</td>
<td>59.8</td>
<td>65.4</td>
<td>67.7</td>
<td>70.2</td>
<td>73.4</td>
</tr>
<tr>
<td>SRC [213]</td>
<td>49.9</td>
<td>60.1</td>
<td>65.0</td>
<td>67.5</td>
<td>69.3</td>
<td>70.9</td>
</tr>
<tr>
<td>DL-COPAR [203]</td>
<td>49.7</td>
<td>58.9</td>
<td>65.2</td>
<td>69.1</td>
<td>71.0</td>
<td>72.9</td>
</tr>
<tr>
<td>K-SVD [2]</td>
<td>51.2</td>
<td>59.1</td>
<td>64.9</td>
<td>68.7</td>
<td>71.0</td>
<td>72.3</td>
</tr>
<tr>
<td>FDDL [221]</td>
<td>52.1</td>
<td>59.8</td>
<td>66.2</td>
<td>68.9</td>
<td>71.3</td>
<td>73.1</td>
</tr>
<tr>
<td>D-KSVD [234]</td>
<td>52.1</td>
<td>60.8</td>
<td>66.1</td>
<td>69.6</td>
<td>70.8</td>
<td>73.1</td>
</tr>
<tr>
<td>LC-KSVD1 [104]</td>
<td>53.1</td>
<td>61.2</td>
<td>66.3</td>
<td>69.8</td>
<td>71.9</td>
<td>73.5</td>
</tr>
<tr>
<td>LC-KSVD2 [104]</td>
<td>53.8</td>
<td>62.8</td>
<td>67.3</td>
<td>70.4</td>
<td>72.6</td>
<td>73.9</td>
</tr>
<tr>
<td>Proposed</td>
<td><strong>53.9</strong></td>
<td><strong>63.1</strong></td>
<td><strong>68.1</strong></td>
<td><strong>71.0</strong></td>
<td><strong>73.3</strong></td>
<td><strong>74.6</strong></td>
</tr>
</tbody>
</table>

Table 10.4: Computation time for training and testing on Caltech-101 database [74].

<table>
<thead>
<tr>
<th>Method</th>
<th>Training (sec)</th>
<th>Testing (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>1474</td>
<td>19.96</td>
</tr>
<tr>
<td>D-KSVD [234]</td>
<td>3196</td>
<td>19.90</td>
</tr>
<tr>
<td>LC-KSVD1 [104]</td>
<td>5434</td>
<td>19.65</td>
</tr>
<tr>
<td>LC-KSVD2 [104]</td>
<td>5434</td>
<td>19.92</td>
</tr>
</tbody>
</table>

more efficiently than LC-KSVD [104] and D-KSVD [234]. This can be verified by comparing the training time of the approaches in Table 10.4. The timings are given for complete training and testing durations for Caltech-101 database, where we used a batch of 30 images per class for training and the remaining images were used for testing. We note that, like all the other approaches, good initialization (using the procedure presented in Section 10.4.4) also contributes towards the computational efficiency of our approach. The training time in the table also includes the initialization time for all the approaches. Note that the testing time of our approach is very similar to those of the other approaches in Table 10.4.
10.5. Experiments

Figure 10.7: Examples images from eight different categories in Fifteen Scene Categories dataset [121].

Table 10.5: Classification accuracy on Fifteen Scene Category dataset [121].

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-SVD [2]</td>
<td>93.60 ± 0.14</td>
</tr>
<tr>
<td>LC-KSVD1[104]</td>
<td>94.05 ± 0.17</td>
</tr>
<tr>
<td>D-KSVD [234]</td>
<td>96.11 ± 0.12</td>
</tr>
<tr>
<td>SRC [213]</td>
<td>96.21 ± 0.09</td>
</tr>
<tr>
<td>DL-COPAR [203]</td>
<td>96.91 ± 0.22</td>
</tr>
<tr>
<td>LC-KSVD2[104]</td>
<td>97.01 ± 0.23</td>
</tr>
<tr>
<td>Proposed</td>
<td>98.73 ± 0.17</td>
</tr>
</tbody>
</table>

10.5.4 Fifteen Scene Category

The Fifteen Scene Category dataset [121] has 200 to 400 images per category for fifteen different kinds of scenes. The scenes include images from kitchens, living rooms and country sides etc. In our experiments, we used the Spatial Pyramid Features of the images, which have been made public by Jiang et al. [104]. In this data, each feature descriptor is a 3000-dimensional vector. Using these features, we performed experiments by randomly selecting 100 training instances per class and considering the remaining as the test instances.

Classification accuracy of the proposed approach is compared with the existing approaches in Table 10.5. The reported values are computed over ten experiments. We set the error tolerance for SRC to $10^{-6}$ and used the parameter settings suggested by Jiang et al. [104] for LC-KSVD1, LC-KSVD2 and D-KSVD. Parameters of DL-COPAR were set as suggested in the original work [203] for the same database. The reported results are obtained by LC for DL-COPAR. Again, the proposed approach shows more accurate results than the existing approaches. The accuracy of the proposed approach is 1.66% more than LC-KSVD2 on the used dataset.
10.5.5 UCF Sports Action

This database comprises video sequences that are collected from different broadcast sports channels (e.g. ESPN and BBC) [172]. The videos contain 10 categories of sports actions that include: kicking, golfing, diving, horse riding, skateboarding, running, swinging, swinging highbar, lifting and walking. Examples from this dataset are shown in Fig. 10.8. Under the common evaluation protocol we performed fivefold cross validation over the dataset, where four folds are used in training and the remaining one is used for testing. Results, computed as the average of the five experiments, are summarized in Table 10.6. For D-KSVD, LC-KSVD1 and LC-KSVD2 we followed [104] for the parameter settings. Again, the value of $10^{-6}$ (along with similar small values) resulted in the best accuracies for SRC.

In the Table, the results for some specific action recognition methods are also included, for instance, Qui et al. [165] and action back feature with SVM [179]. These results are taken directly from [220] along the results of DLSI [166], DL-COPAR [203] and FDDL [221]. Following [179], we also performed leave-one-out cross validation on this database for the proposed approach. Our approach achieves 95.7% accuracy under this protocol, which is 0.7% better than the state-of-the-art results claimed in [179].

10.6 Discussion

In our experiments, we used large values for $\lambda_{c_o}$, because this parameter represents the precision of the white noise distribution in the samples. The datasets used in our experiments are mainly clean in terms of white noise. Therefore, we achieved the best performance with $\lambda_{c_o} \geq 10^6$. In the case of noisy data, this parameter value can be adjusted accordingly. For UCF sports action dataset $\lambda_{c_o} = 10^9$ gave

---

5The results of DL-COPAR [203] and FDDL [221] are taken directly from the literature because the optimized parameter values for these algorithms are not previously reported for this dataset. Our parameter optimization did not outperform the reported accuracies.
Table 10.6: Classification rates on UCF Sports Action [172].

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy %</th>
<th>Method</th>
<th>Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qiu et al. [165]</td>
<td>83.6</td>
<td>LC-KSVD2 [104]</td>
<td>91.5</td>
</tr>
<tr>
<td>D-KSVD [234]</td>
<td>89.1</td>
<td>DLSI [166]</td>
<td>92.1</td>
</tr>
<tr>
<td>LC-KSVD1 [104]</td>
<td>89.6</td>
<td>SRC [213]</td>
<td>92.7</td>
</tr>
<tr>
<td>DL-COPAR [203]</td>
<td>90.7</td>
<td>FDDL [221]</td>
<td>93.6</td>
</tr>
<tr>
<td>Sadanand [179]</td>
<td>90.7</td>
<td>LDL [220]</td>
<td>95.0</td>
</tr>
<tr>
<td>Proposed</td>
<td><strong>95.1</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

the best results because less number of training samples were available per class. It should be noted that the value of \( \lambda \) increases as a result of Bayesian inference with the availability of more clean training samples. Therefore, we adjusted the precision parameter of the prior distribution to a larger value for UCF dataset. Among the other parameters, \( c_o \) to \( f_o \) were fixed to \( 10^{-6} \). Similar small non-negative values can also be used without affecting the results. This fact can be easily verified by noticing the large values of the other variables involved in Equations (10.12) and (10.13), where these parameters are used. With the above mentioned parameter settings and the initialization procedure presented in Section 10.4.4, the Gibbs sampling process converges quickly to the desired distributions and the correct number of dictionary atoms, i.e. \( |K| \). In Fig. 10.9, we plot the value of \( |K| \) as a function of Gibbs sampling iterations during dictionary training. It can be easily seen that the first few iterations of the Gibbs sampling process were generally enough to converge to the correct size of the dictionary. However, it should be mentioned that this fast convergence also owes to the initialization process adopted in this work. In our experiments, while sparse coding a test instance over the learned dictionary, we consistently used the sparsity threshold of 50 for all the datasets except for the UCF [172], for which this parameter was set to 40 because of the smaller dictionary resulting from less training samples. In all the experiments, this parameter value was also kept the same for K-SVD, LC-KSVD1, LC-KSVD2 and D-KSVD.

It is worth mentioning that our model in Eq. (10.6) can also be exploited for simultaneously learning the dictionary and the classifier. Therefore, we also explored this alternative for our model. For that, we used the matrix \( [X; H] \in \mathbb{R}^{(m+C)\times N} \) as the training data, where ‘;’ denotes the vertical concatenation of the matrices and \( H \in \mathbb{R}^{C\times N} \) is created by arranging the vectors \( h_c \) for the training samples. For such training data, the basis inferred by our model can be seen as \( [\Phi; W] \in \mathbb{R}^{(m+C)\times |K|} \).
Figure 10.9: Size of the inferred dictionary, i.e. $|\mathcal{K}|$, as a function of the Gibbs sampling iterations. The plots show first hundred iterations for each dataset.

This approach of joint dictionary and classifier training is inspired by D-KSVD [234] and LC-KSVD [104], that exploit the K-SVD model [2] in a similar fashion. Since that model is for unsupervised dictionary learning, its extension towards supervised training by the joint learning procedure yielded improved classification performance for D-KSVD [234] and LC-KSVD [104], as compared to K-SVD [2]. However, the joint training procedure did not have a similar effect on our approach. In fact, in our experiments, the approach presented in Section 10.4 achieved an average percentage gain of 0.93, 0.61, 0.32 and 0.50 for AR database [140], Extended YaleB [77], Caltech-101 [74] and Fifteen Scene Category database [121], respectively over the joint learning procedure. This consistent performance gain is an expected phenomenon for our approach. As mentioned in Section 10.4.2, for computational purposes, we assume isotropic precision Gaussians over the basis vectors in our model. By jointly learning the classifier with the dictionary, its weights must also follow the same distributions, as followed by the dictionary atoms. This assumption is restrictive, which results in a slight degradation of the classification accuracy. The separate learning procedures for the dictionary and the classifier remove this restrictive assumption while keeping the inference process efficient.

10.7 Conclusion

We proposed a non-parametric Bayesian approach for learning discriminative dictionaries for sparse representation of data. The proposed approach employs a truncated Beta process to infer a discriminative dictionary and sets of Bernoulli distributions associating the dictionary atoms to the class labels of the training data. The said association is adaptively built during Bayesian inference and it signifies the selection probabilities of dictionary atoms in the expansion of class-specific data. The inference process also results in computing the correct size of the dictionary. For learning the discriminative dictionary, we presented a hierarchical Bayesian model and the corresponding inference equations for Gibbs sampling. The
proposed model is also exploited in learning a linear classifier that finally classifies the sparse codes of a test instance that are learned using the inferred discriminative dictionary. The proposed approach is evaluated for classification using five different databases of human face, human action, scene category and object images. Comparisons with state-of-the-art discriminative sparse representation approaches show that the proposed Bayesian approach consistently outperforms these approaches and has computational efficiency close to the most efficient approach.

Whereas its effectiveness in terms of accuracy and computation is experimentally proven in this work, there are also other key advantages that make our Bayesian approach to discriminative sparse representation much more appealing than the existing optimization based approaches. Firstly, the Bayesian framework allows us to learn an ensemble of discriminative dictionaries in the form of probability distributions instead of the point estimates that are learned by the optimization based approaches. Secondly, it provides a principled approach to estimate the required dictionary size and we can associate the dictionary atoms and the class labels in a physically meaningful manner. Thirdly, the Bayesian framework makes our approach inherently an online technique. Furthermore, the Bayesian framework also provides an opportunity of using domain/class-specific prior knowledge in our approach in a principled manner. This can prove beneficial in many applications. For instance, while classifying the spectral signatures of minerals on pixel and sub-pixel level in remote-sensing hyperspectral images, the relative smoothness of spectral signatures [6] can be incorporated in the inferred discriminative bases. For this purpose, Gaussian Processes [167] can be used as a base measure for the Beta Process. Adapting the proposed approach for remote-sensing hyperspectral image classification is also our future research direction.
Joint Bayesian Dictionary and Classifier Learning

Abstract

We propose to jointly learn a Bayesian dictionary with a linear classifier using coupled Beta Processes. Our approach infers separate probability distributions over the dictionary and the classifier such that they are associated with the class-specific training data by a common set of Bernoulli distributions. In a Beta Process, the Bernoulli distributions signify the frequency of dictionary atom usage in data representation. Our model employs separate (sets of) Bernoulli distributions to represent data from different classes, rendering the dictionary discriminative. Representations of test samples over the dictionary are accurately classified by the classifier due to the coupling between the dictionary and the classifier. We derive the Gibbs Sampling equations for our joint representation model and test our approach for face, object and action recognition tasks to establish its effectiveness.

Keywords: Classification, joint dictionary learning, coupled Beta Process.

11.1 Introduction

Dictionary Learning [195] is a well-known signal representation technique, employed in compressive sensing [67], image restoration [71] and morphological component analysis [29]. A dictionary is a set of basis vectors (a.k.a. atoms), learned to represent data. More recently, this technique has shown great potential for multiple classification tasks [104], [221], [220], [132], [234]. Dictionaries learned for classification (a.k.a. discriminative dictionaries) not only represent the training data from different classes accurately, they also render their representations easily classifiable by a suitable classifier.

Generally, discriminative dictionary learning approaches either restrict subsets of the dictionary atoms to represent data of specific classes only [166], [133]; or they force the representations of the data over the entire dictionary to become discriminative [104], [234]. In some instances, a dictionary is also learned as a concatenation of class-specific atoms and atoms to jointly represent the data from all the classes [203], [166]. In any case, the relationship between the dictionary atoms and the class labels remains the key for effective discriminative dictionaries [117]. Nevertheless, adaptive

\footnote{Submitted to the Annual Conf. on Neural Information Processing Systems (NIPS), 2016.}
learning of this relation is still a largely open research problem [220]. Subsequently, tailoring a classifier to this learned relation for effective classification also remains unaddressed.

In this work, we present a non-parametric Bayesian approach to fill in the aforementioned research gaps. We propose a finite Beta Process [157] based representation model to infer a dictionary whose atoms are related to different class data under different sets of Bernoulli distributions. These distributions are adaptively learned in our approach with Bayesian inference and they signify the frequency of the dictionary atom usage in data representations of the respective classes. Our model couples a second Beta Process with the dictionary learning process, that employs its own base measure for learning the classifier, but uses the same Bernoulli distributions as used by the dictionary learning process. Since the test and training samples of the same class are likely to use the relevant dictionary atoms with similar frequency, incorporating the dictionary atom popularity for different classes into the classifier (using a joint learning process) results in accurate classification.

We derive Gibbs Sampling equations for our model and test our approach on popular benchmark datasets for face recognition [77], object classification [74] and action recognition [172]. Experiments show that our approach consistently improves the classification accuracy over the existing state-of-the-art dictionary learning/sparse representation based approaches, resulting in up to 24% reduction in the error rate. The implementation of the approach will be made public.

11.2 Problem Settings and Preliminaries

Let the $i^{th}$ training sample of the $c^{th}$ class be expressed as $y_c^i = \Phi \alpha_c^i + y_\epsilon^i$, where $\Phi \in \mathbb{R}^{L \times |K|}$ is the dictionary, $\alpha_c^i \in \mathbb{R}^{|K|}$ is the representation of the sample over the dictionary and $y_\epsilon^i \in \mathbb{R}^L$ denotes the Gaussian noise. The $k^{th}$ atom $\phi_k$ of the dictionary is indexed in the set $K = \{1, ..., K\}$, whose cardinality $|K|$ is unknown beforehand. The training samples of the $c^{th}$ class are indexed in the set $I_c$; and $\sum_{c=1}^{C} |I_c| = N$, where $C$ denotes the total number of classes. We follow the convention that absence of the superscript ‘c’ implies that no distinction is being made between different classes for the variable under consideration. For instance, $y_\epsilon^i$ is not annotated with ‘c’ because the training samples from all the classes are considered to have common noise statistics.$^1$

---

$^1$We avoid unnecessary modeling complexity with this simplification. In our experiments, the classification performance did not change significantly when different noise statistics were used for each class.
A typical dictionary learning approach solves the following optimization problem to learn $\Phi$:

$$< \Phi, \alpha_i > = \min_{\Phi, \alpha_i} ||y_i - \Phi \alpha_i||^2_2 \quad \text{s.t.} \quad \forall i, ||\alpha_i||_p \leq t,$$

where $||.||_p$ denotes the $\ell_p$-norm and ‘$t$’ is a pre-defined constant. Using thus computed $\alpha_i$, we can learn a linear classifier $\Psi \in \mathbb{R}^{C \times |K|}$ by solving:

$$< \Psi > = \min_{\Psi} \sum_{i=1}^{N} L\{h_i, f(\alpha_i, \Psi)\} + \lambda ||\Psi||^2_F,$$

where $L$ denotes the loss function, $\lambda$ is the regularizer, $h_i \in \mathbb{R}^C$ is the class label for $y_i$ and $f(.)$ results in the predicted label. In these settings, a test sample can be classified by first computing its representation $\hat{\alpha}$ over $\Phi$ and then classifying $\hat{\alpha}$ using $\Psi$. Nevertheless, since $\Phi$ is learned in an unsupervised fashion, the classification performance is expected to remain sub-optimal. To overcome this issue, Jiang et al. [104] and Zhang and Li [234] proposed to learn the classifier jointly with the dictionary in a supervised manner. However, the results of their approaches strongly depend on the used dictionary size, because the accuracy of data representation actively depends on this parameter [213]. Moreover, their approaches pre-fix the relationship between the dictionary atoms and the class labels, which is not an attractive machine learning strategy.

Paisley and Carin [157] proposed a Beta Process that can be used to learn $\Phi$ non-parametrically, thereby automatically inferring the appropriate dictionary size. With its base measure $h_0$ and parameters $a, b > 0$, a finite representation of the Beta Process is given as [157]:

$$h = \sum_{k \in K} \pi_k \delta_{\varphi_k}(\varphi); \quad \pi_k \sim \text{Beta}\left(\frac{a}{K}, \frac{b(K-1)}{K}\right); \quad \varphi_k \sim h_0,$$

where $\delta_{\varphi_k}(\varphi) = 1$ when $\varphi = \varphi_k$ and 0 otherwise. A draw $h$ from the process is a set of $|K|$ probabilities $\pi_k \in K$, each associated with a $\varphi_k \in K$ that is drawn i.i.d. from the base measure. Considering $\pi_k$ to be a Bernoulli distribution parameter, we can use $h$ to draw a binary vector $z \in \mathbb{R}^{|K|}$ such that its $k^{th}$ coefficient follows $\text{Bernoulli}(\pi_k)$.

Drawing $N$ binary vectors $z_i \in \{1,...,N\}$ under $B = \{\text{Bernoulli}(\pi_k) : k \in K\}$ using the Beta Process, the training data can be factorized as: $y_i \approx \Phi z_i$, $\forall i$, where the atoms $\varphi_k$ of the dictionary $\Phi$ are the base measure draws. In the limit $K \to \infty$, the number of the non-zero elements in $z_i$ is itself a draw from $\text{Poisson}(\frac{a}{b})$ [157], which adjusts the dictionary size non-parametrically. Notice that the set $B$ associates
the dictionary atoms to the training data through the vectors $z_i$ such that, the $k^{th}$ distribution in $B$ signifies the frequency of the $k^{th}$ atom usage in the data expansion. Our approach exploits the existence of this set in the Beta Process to render the non-parametrically inferred dictionary discriminative. This also results in adaptive learning of the relationship between the atoms and the class labels, which is further exploited in the joint inference of the classifier with the dictionary.

11.3 Proposed Approach

We propose to learn a Bayesian dictionary jointly with a linear classifier in a supervised manner, using two coupled Beta Processes [157]. To learn the dictionary, we use separate draws of a finite Beta Process for each class, however, under a common base measure. That is, we learn a common dictionary to represent the data from all the classes, but the data from each class is associated with the dictionary through a distinct set of Bernoulli distributions. We factorize the $c^{th}$ class data by drawing $|I_c|$ binary vectors $z_{ci} \in I_c$ from the $c^{th}$ Bernoulli distributions set. To jointly learn the classifier with the dictionary, we employ a second Beta Process that uses the same sets of the Bernoulli distributions as used by the dictionary learning process, but it uses a different base measure to draw the classifier parameters. The Bernoulli distributions couple the two Beta Processes for simultaneous inference of an effective dictionary and an accurate classifier. We use Gibbs Sampling\(^2\) to perform the inference.

11.3.1 The Model

Representing the data as mere binary combinations of the basis vectors is restrictive. Therefore, we let $y_c^i = \Phi(z_{ci} \odot s_{ci}) + \epsilon_i$, where $s_{ci} \in \mathbb{R}^{|K|}$ is the weight vector and $z_{ci} \odot s_{ci} = \alpha_{ci};$ where $\odot$ denotes the element-wise product. This factorization is possible under a weighted Beta Process, as shown below. For each $y_c^i$, we represent its label vector as $h_c^i = \Psi(z_{ci} \odot t_{ci}) + h^i$ and use the following joint hierarchical Bayesian model to learn the dictionary and the classifier:

\[
\forall i \in I_c \text{ and } \forall k \in K = \{1, ..., K\}:
\]

\(^2\)A variational algorithm was developed for Beta Process in [157]. Later, Zhou et al. [238] showed Gibbs Sampling to be equally effective for Bayesian dictionary learning. As the latter is intuitively related to the optimization based algorithms for learning discriminative dictionaries, we also use Gibbs Sampling.
\[ y^c_i = \Phi(z^c_i \odot s^c_i) + \epsilon^c_i \quad \quad h^c_i = \Psi(z^c_i \odot t^c_i) + \epsilon^c_i \quad (11.4) \]

\[ z^c_{ik} \sim \text{Bernoulli}(z^c_{ik} | \pi^c_k) \]

\[ \pi^c_k \sim \text{Beta}(\pi^c_k | a_o / K, b_o (K - 1) / K) \]

\[ s^c_{ik} \sim \mathcal{N}(s^c_{ik} | 0, 1 / \lambda^c_o) \quad \quad t^c_{ik} \sim \mathcal{N}(t^c_{ik} | 0, 1 / \lambda^c_o) \]

\[ \varphi_k \sim \mathcal{N}(\varphi_k | 0, \Lambda^{-1}_c) \quad \quad \psi_k \sim \mathcal{N}(\psi_k | 0, \Lambda^{-1}_c) \]

\[ \epsilon^c_i \sim \mathcal{N}(\epsilon^c_i | 0, 1 / \lambda^c_o I_L) \quad \quad h^c_i \sim \mathcal{N}(h^c_i | 0, 1 / \lambda^c_h I_C), \]

where, \( \lambda \) and \( \Lambda \) represent the Gaussian distribution precision parameters, \( \psi_k \in \mathbb{R}^C \) is the \( k^{th} \) column of \( \Psi \), \( I_Q \) is an identity matrix in \( \mathbb{R}^{Q \times Q} \), \( 0 \) is a vector of zeros with an appropriate dimension and the subscript ‘o’ indicates that the associated parameter belongs to a prior distribution.

In the above model, both \( y^c_i \) and \( h^c_i \) use the same \( z^c_i \), whose \( k^{th} \) coefficient \( z^c_{ik} \) is drawn from a Bernoulli distribution, with a conjugate Beta prior. The coefficients of the weight vectors \( s^c_i, t^c_i \in \mathbb{R}^{K|} \) are drawn from separate Gaussian distributions. Similarly, the dictionary atoms and the classifier parameters are drawn from distinct multivariate Gaussians. This allows the representations of \( y^c_i \) and \( h^c_i \); as well as the dictionary and classifier to be accurate, yet remain coupled. The model is also flexible to allow the additive noise/modeling error for \( y^c_i \) and \( h^c_i \) to be the samples of different distributions. We further place the following non-informative Gamma hyper-priors on the precision parameters of the distributions: \( \lambda^c_o, \Lambda^c_o \sim \text{Gam}(c_o, d_o) \) and \( \lambda_y, \lambda_h \sim \text{Gam}(c_o, f_o) \). The graphical representation of the model and the analytical expression for the underlying joint probability distribution are provided in Appendix E.1 and E.2, respectively.

According to Eq. (11.4), the covariances of \( y^c_i \) and \( h^c_i \) can be given as \( \mathbb{E}[y^c_i y^c_i^\top] = \frac{aK}{a + b(K - 1)} \frac{\Lambda^c_o}{\lambda^c_o} + I_c \) and \( \mathbb{E}[h^c_i h^c_i^\top] = \frac{aK}{a + b(K - 1)} \frac{\Lambda^c_o}{\lambda^c_o} + I_c \), respectively. Recall, that \( K \) signifies the number of vectors \( \varphi_k \) or \( \psi_k \) in our settings. By letting \( K \rightarrow \infty \), \( \mathbb{E}[y^c_i y^c_i^\top] \rightarrow \frac{a}{b} \frac{\Lambda^c_o}{\lambda^c_o} + I_c \) and \( \mathbb{E}[h^c_i h^c_i^\top] \rightarrow \frac{a}{b} \frac{\Lambda^c_o}{\lambda^c_o} + I_c \); which shows that the joint model remains well defined in the infinite limit. However, it is important to notice that when \( K \) is infinite, the model restricts the samples from different classes to use different sets of basis vectors in their representations. Impliedly, the dictionary becomes a concatenation of class-specific dictionary atoms, as in [166], [133]. On the other hand, for a practically large value of \( K \), which is valid for the finite Beta Process used in this work, the model allows different classes to share the dictionary atoms. As per the discriminative nature of the dictionary, the atom sharing is expected to be such that along popular groups of dictionary atoms for a given class,
11.3. Proposed Approach

Figure 11.1: Dictionary atom usage for the data representation of three classes of Caltech-101 [74]: For classes 10, 50 and 90, the Bernoulli distribution parameters \( \pi^c_k \) are plotted against the indices of the arranged atoms. A larger \( \pi^c_k \) indicates a more frequent use of the atom in data representation. Due to the discriminative nature of the dictionary, distinct clusters of popular atoms appear. Due to sharing of the atoms by different classes, non-zero values also appear far from the main clusters. The shown relationships between the atoms and the class labels are learned adaptively by our approach.

Few other atoms may also have non-zero probability of selection for representing that class. Fig. 11.1 verifies this behavior of our model for an actual object recognition task. It plots the inferred parameters \( \pi^c_k \in \{10, 50, 90\} \) of the Bernoulli distributions for an arranged dictionary. For each class, a cluster of large \( \pi^c_k \) values is observable at a distinct location, indicating the discriminative character of the inferred dictionary. However, few non-zero values can also be observed far from each cluster. These values are indicative of the atom sharing between the classes.

In the above mentioned covariances of \( y^c_i \) and \( h^c_i \), the fraction \( \frac{a}{a+b(K-1)} \) appears due to \( z^c_i \) in Eq. (11.4). Ignoring this fraction and comparing the remaining expressions with those when \( K \to \infty \) shows that the value \( \frac{a}{b} \) signifies the expected number of the basis vectors required to represent \( y^c_i \) or \( h^c_i \) in our model. This is similar to the original model of the Beta Process [157], except that the data under consideration is class-specific in our approach. This observation demonstrates the non-parametric character of our model. To take advantage of which, we apply our model with \( K = 1.25 \times N \) for the initialization and let the inference process determine the correct dictionary size, i.e. \( |K| \).

11.3.2 Inference

We perform Gibbs Sampling for inference over the model. Due to the conjugacy of the distributions, all the sampling equation are derived analytically. Please also see Appendix E.3 for further details on the derivation. To simplify the sampling procedure, below, we let \( \Lambda_{\varphi_a} = I_L/\lambda_{\varphi_a} \) and \( \Lambda_{\psi_a} = I_C/\lambda_{\psi_a} \). In our experiments, this
simplification considerably reduced the computational time, without significantly affecting the classification performance.

**Sample $\varphi_k$:** According to the proposed model, we can write the following regarding the posterior probability distribution $p(\varphi_k | -)$ over the $k^{th}$ dictionary atom:

$$p(\varphi_k | -) \propto \prod_{i=1}^{N} \mathcal{N}(y_{i \varphi_k} | \varphi_k(z_{ik} \cdot s_{ik}), \lambda_{y_{ik}}^{-1} I_L) \mathcal{N}(\varphi_k | 0, \lambda_{\varphi_k}^{-1} I_L),$$

where, $y_{i \varphi_k} = y_i - \Phi(z_i \otimes s_i) + \varphi_k(z_{ik} \cdot s_{ik})$ denotes the contribution of the $k^{th}$ dictionary atom in approximating $y_i$. Note the absence of the superscript ‘c’. It indicates that the atom is being treated alike for all the classes and is being updated using the complete training data. Exploiting the conjugacy of the Gaussian distributions, $\varphi_k$ can be sampled from $\mathcal{N}(\varphi_k | \mu_k, \lambda_{\varphi_k}^{-1} I_L)$, where:

$$\lambda_{\varphi} = \lambda_{\varphi_0} + \lambda_{y_{ik}} \sum_{i=1}^{N} (z_{ik} \cdot s_{ik})^2, \quad \mu_k = \lambda_{y_{ik}} \lambda_{\varphi_k}^{-1} \sum_{i=1}^{N} (z_{ik} \cdot s_{ik}) y_{i \varphi_k}.$$

**Sample $\psi_k$:** With the same reasoning as for sampling the dictionary atoms, we sample $\psi_k$ from $\mathcal{N}(\psi_k | \mu_k, \lambda_{\psi_k}^{-1} I_C)$, such that:

$$\lambda_{\psi} = \lambda_{\psi_0} + \lambda_{h_{ik}} \sum_{i=1}^{N} (z_{ik} \cdot t_{ik})^2, \quad \mu_k = \lambda_{h_{ik}} \lambda_{\psi_k}^{-1} \sum_{i=1}^{N} (z_{ik} \cdot t_{ik}) h_{i \psi_k}.$$

**Sample $z_{ik}^c$:** Once the dictionary and the classifier have been sampled, we sample $z_{ik}^c$ using their updated versions. The posterior probability distribution over $z_{ik}^c$ can be expressed as, $\forall i \in \mathcal{I}$, $\forall k \in \mathcal{K}$:

$$p(z_{ik}^c | -) \propto \mathcal{N}(y_{i \varphi_k}^c | \varphi_k(z_{ik} \cdot s_{ik}^c), \lambda_{y_{ik}}^{-1} I_L) \mathcal{N}(h_{i \psi_k}^c | \psi_k(z_{ik} \cdot t_{ik}^c), \lambda_{h_{ik}}^{-1} I_C) \text{Bernoulli}(z_{ik}^c | \pi_{k_{ha}}).$$

Based on the above expression, it can be shown that $z_{ik}^c$ should be sampled from

$$z_{ik}^c \sim \text{Bernoulli}(\frac{\pi_{k_{ha}}^c \xi_1 \xi_2}{1 - \pi_{k_{ha}} + \xi_1 \xi_2 \pi_{k_{ha}}}),$$

where

$$\xi_1 = \exp\left( -\frac{\lambda_{y_{ik}}}{2} (\varphi_k^T \varphi_k s_{ik}^2 - 2 s_{ik}^c y_{i \varphi_k}^c \varphi_k) \right) \quad \text{and} \quad \xi_2 = \exp\left( -\frac{\lambda_{h_{ik}}}{2} (\psi_k^T \psi_k t_{ik}^2 - 2 t_{ik} h_{i \psi_k}^c \psi_k) \right).$$

**Sample $s_{ik}^c$:** We can write the following regarding the posterior probability distribution over $s_{ik}^c$:

$$p(s_{ik}^c | -) \propto \mathcal{N}(y_{i \varphi_k}^c | \varphi_k(z_{ik} \cdot s_{ik}^c), \lambda_{y_{ik}}^{-1} I_L) \mathcal{N}(s_{ik}^c | 0, \lambda_{s_{ik}}^{-1}).$$

Exploiting the conjugacy between the distributions, $s_{ik}^c$ can be sampled from the distribution $\mathcal{N}(s_{ik}^c | \mu_s, \lambda_s^{-1})$, where:

$$\lambda_s = \lambda_{s_0} + \lambda_{y_{ik}} z_{ik}^2 \varphi_k \varphi_k, \quad \mu_s = \lambda_s^{-1} \lambda_{y_{ik}} z_{ik}^c \varphi_k y_{i \varphi_k}^c.$$
Sample $t_{ik}^c$: Similarly, we can sample $t_{ik}^c$ from $\mathcal{N}(t_{ik}^c | \mu_t, \lambda_t^{-1})$, where:

$$
\lambda_t = \lambda_{t_0} + \lambda_{b_o} z_{ik}^c \psi_k^\top \psi_k, \quad \mu_t = \lambda_t^{-1} \lambda_{b_o} z_{ik}^c \psi_k^\top h_k^c.
$$

Sample $\pi_k^c$: The posterior probability distribution over $\pi_k^c$ can be written as follows:

$$
\begin{align*}
p(\pi_k^c | -) & \propto \prod_{i \in I_c} \text{Bernoulli}(z_{ik}^c | \pi_k^c) \text{Beta}(\pi_k^c | a_o/K, b_o(K - 1)/K), \\
& \propto \text{Beta} \left( \frac{a_o}{K} + \sum_{i=1}^{|I_c|} z_{ik}^c, \frac{b_o(K - 1)}{K} + |I_c| - \sum_{i=1}^{|I_c|} z_{ik}^c \right).
\end{align*}
$$

Hence, we sample $\pi_k^c$ from the above mentioned Beta probability distribution. Note that, our sampling process also infers the correct dictionary size $|K|$ by monitoring the sampled values of $\pi_k^c, \forall c$. In the light of Lemma 7 below, we drop out the $\varphi_k$ and $\psi_k$ in the subsequent sampling iterations for whom $\sum_{c=1}^C \pi_k^c \rightarrow 0$ in the current iteration.

**Lemma 7.** If $\sum_{c=1}^C \pi_k^c \rightarrow 0$ in a given Gibbs Sampling iteration, $\mathbb{E}[\pi_k^c] \approx 0$ for the later iterations when $a_o, b_o < |I_c| \ll K$.

**Proof:** When $\sum_{c=1}^C \pi_k^c \rightarrow 0$ in an iteration, $\sum_{i=1}^{|I_c|} z_{ik}^c \rightarrow 0, \forall c$ in the next iteration. This results in approximating the posterior distribution over $\pi_k^c$ as the probability distribution $\text{Beta}(\pi_k^c | a_o/K, b_o(K - 1)/K + |I_c|)$, for which, $\mathbb{E}[\pi_k^c] = \frac{a_o}{a_o + b_o(K - 1) + K|I_c|}$. For $0 < a_o, b_o < |I_c| \ll K$, $\mathbb{E}[\pi_k^c] \approx 0$.

Sample $\lambda_s^c$: To compute $\lambda_s^c$, we treat $s_{ik}^c$ for all the dictionary atoms simultaneously (we do the same for $\lambda_t^c$ below). We consider $s_i^c \in \mathbb{R}^{|K|}$ to be drawn from a multivariate Gaussian with isotropic precision. This simplification allows us to efficiently infer the posterior distribution over $\lambda_s^c$. The posterior over $\lambda_s^c$ can be given as:

$$
p(\lambda_s^c | -) \propto \prod_{i \in I_c} \mathcal{N}(s_i^c | 0, 1/\lambda_s^c \mathbf{1}_{|K|}) \text{Gam}(\lambda_s^c | c_o, d_o).
$$

Exploiting the conjugacy between the Gaussian and the Gamma distributions, we sample $\lambda_s^c$ as:

$$
\lambda_s^c \sim \text{Gam} \left( \frac{|I_c||K|}{2} + c_o, \frac{1}{2} \sum_{i=1}^{|I_c|} ||s_i^c||_2^2 + d_o \right).
$$

Sample $\lambda_t^c$: Correspondingly, we also sample $\lambda_t^c$ from the Gamma probability distribution mentioned above, with $t_i^c$ replacing $s_i^c$ in the expression.
Sample $\lambda_y$: The posterior probability distribution over $\lambda_y$ can be written as:

$$p(\lambda_y|\cdot) \propto \prod_{i=1}^{N} \mathcal{N}(y_i|\Phi(z_i \odot s_i), \lambda_{y_i}^{-1} I_L) \text{Gam}(\lambda_y|e_o, f_o).$$

Again, we do not use the superscript ‘c’ because $\lambda_y$ is sampled utilizing the training data from all the classes. Similar to the case of $\lambda^c$, we can show that $\lambda_y$ should be sampled as follows:

$$\lambda_y \sim \text{Gam} \left( \frac{LN}{2} + e_o, \frac{1}{2} \sum_{i=1}^{N} ||y_i - \Phi(z_i \odot s_i)||_2^2 + f_o \right).$$

Sample $\lambda_h$: Analogously, $\lambda_h$ is sampled using the following Gamma distribution:

$$\lambda_h \sim \text{Gam} \left( \frac{CN}{2} + e_o, \frac{1}{2} \sum_{i=1}^{N} ||h_i - \Psi(z_i \odot t_i)||_2^2 + f_o \right).$$

As a result of the sampling process we infer posterior distributions over the dictionary atoms and the classifier parameters. We sample these distributions to obtain $\Phi$ and $\Psi$. To classify a test sample, we first compute its representation $\hat{\alpha}$ over $\Phi$ and then predict the label by classifying $\hat{\alpha}$ with $\Psi$. We use the Orthogonal Matching Pursuit (OMP) algorithm [33] to compute $\hat{\alpha}$. A Beta Process tends to make a representation vector sparse by forcing most of its coefficients to become exactly zero, similar to OMP. Therefore, OMP is a natural choice for computing $\hat{\alpha}$ in our approach.

To start the sampling process, we initialize the dictionary atoms by randomly selecting samples from the training data with replacement. We compute the sparse codes of the training data over the initial $\Phi$ with OMP and use them as the initial values of $s^c_i$ and $t^c_i$. The vectors $z^c_i$ are computed by replacing the non-zero coefficients of the initial $s^c_i$ with ones. The initial value of $\Psi$ is computed with the help of ridge regression, using $t^c_i$ and the training labels. This initialization procedure is inspired by the popular discriminative dictionary learning approaches [104], [221] and [234].

### 11.4 Experiments

We evaluated our approach for face, object, scene, and action recognition tasks using benchmark data sets. The performance is compared with the Label Consistent K-SVD (LC-KSVD) [104], Sparse Representation based Classification (SRC) [213], Discriminative Bayesian Dictionary Learning (DBDL) [7], Discriminative K-SVD (D-KSVD) [234], Fisher Discrimination Dictionary Learning (FDDL) [221] and the
Dictionary Learning based on Commonalities and Particularities of the data (DL-COPAR) [203]. These are the state-of-the-art methods in the area of discriminative dictionary learning/sparse representation. We also include the results of an LC-KSVD variant LC-KSVD1, that computes the classifier separately from the dictionary [104].

We used the author-provided implementations of all the methods, except for SRC and D-KSVD. We implemented SRC using the SPAMS toolbox [136], whereas the public code of LC-KSVD [104] was modified for D-KSVD, as recommended by Jiang et al [104]. For all the methods that use OMP to compute the sparse codes, we used the implementation of OMP provided by Elad et al. [177]. In our experiments, all the approaches use the same training and test data. The reported results have been computed after careful optimization of the parameters for each method using cross-validation. We followed the guidelines provided in the original works for selecting the parameter ranges. Discussion on parameter value selection of the proposed approach is provided in Section 11.5. Experiments were conducted on an Intel processor at 3.4 GHz, with 16 GB RAM.

11.4.1 Face recognition

We experimented with two commonly used face databases: Extended YaleB [77] and the AR database [140].

**Extended YaleB database:** This database comprises 2,414 images of 38 subjects. The images have large variations in terms of illumination conditions and expressions for each subject. To use the images in our experiments, we first created 504-dimensional random face features [213] from the 192 × 168 cropped face images. For each experiment, we randomly selected 15 features per subject for training and the remaining samples were used for testing. We conducted 10 experiments by randomly selecting the training and testing samples. The means±std.dev of the resulting recognition rates are reported in Table 11.2. We abbreviate the proposed approach as JBDC for Joint Bayesian Dictionary and Classifier learning.

The proposed approach resulted in 11.8% reduction in the error rate in Table 11.2. This reduction is achieved over a recently proposed Bayesian discriminative dictionary learning technique [7]. In our opinion, the better performance of our approach over DBDL is attributed to the stronger coupling between the dictionary and the classifier, and to the ability of JBDC to exploit the relation between the class labels and the factors for the dictionary and the classifier alike. The recognition time of JBDC is comparable to those of the efficient approaches. The low
Table 11.1: Face recognition results on Extended YaleB database [77]. Results are averaged over ten experiments. The time is given for classifying a single test sample.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy %</th>
<th>Average Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DL-COPAR [203]</td>
<td>86.47 ± 0.69</td>
<td>31.11</td>
</tr>
<tr>
<td>LC-KSVD1 [104]</td>
<td>87.76 ± 0.60</td>
<td>0.61</td>
</tr>
<tr>
<td>LC-KSVD [104]</td>
<td>89.73 ± 0.59</td>
<td>0.60</td>
</tr>
<tr>
<td>D-KSVD [234]</td>
<td>89.77 ± 0.57</td>
<td>0.61</td>
</tr>
<tr>
<td>SRC [213]</td>
<td>89.71 ± 0.45</td>
<td>50.19</td>
</tr>
<tr>
<td>FDDL [221]</td>
<td>90.01 ± 0.69</td>
<td>42.82</td>
</tr>
<tr>
<td>DBDL [7]</td>
<td>91.09 ± 0.59</td>
<td>1.07</td>
</tr>
<tr>
<td>JBDC (Proposed)</td>
<td>92.14 ± 0.52</td>
<td>1.02</td>
</tr>
</tbody>
</table>

recognition time owes to the joint learning of the classifier along the dictionary. The dictionary/classifier size inferred by JBDC is generally smaller than the dictionary size computed by DBDL which also gives a slight computational advantage to our approach over DBDL. However, the final dictionary size of JBDC is generally larger than the optimal dictionary sizes for D-KSVD and LC-KSVD, which benefits these approaches computationally. Nevertheless, the accuracy of the proposed approach remains significantly better than these approaches. In our experiments, the average dictionary size computed by JBDC was 567 atoms, whereas this value was 574 for DBDL. LC-KSVD and D-KSVD used 375 dictionary atoms, which resulted in their best performance.

**AR face database:** This database consists of over 4,000 face images of 126 subjects. For each subject, 26 images are taken during two different sessions such that they have large variations in facial disguise, illumination and expressions. We projected 165 × 120 cropped face images onto 540-dimensional vectors using a random projection matrix, thereby extracting Random-Face features [213]. Following a common evaluation protocol, we selected a subset of 2,600 images of 50 male and 50 female subjects from the database. For each subject, we used 7 randomly selected images for training and the remaining images were used for testing. Results of our recognition experiments are summarized in Table 11.2. Similar to the Extended YaleB data set, the proposed approach is also able to generally perform better than the existing approaches on AR database. On average, as compared to 705 dictionary atoms learned by DBDL, JBDC inferred 697 atoms for the training data.
Table 11.2: Face recognition on AR database [140]. Results are averaged over ten experiments. The time is for a single test sample.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy %</th>
<th>Average Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DL-COPAR [203]</td>
<td>83.29 ± 1.23</td>
<td>36.49</td>
</tr>
<tr>
<td>SRC [213]</td>
<td>84.60 ± 1.37</td>
<td>59.91</td>
</tr>
<tr>
<td>LC-KSVD1 [104]</td>
<td>84.61 ± 1.44</td>
<td>0.91</td>
</tr>
<tr>
<td>LC-KSVD [104]</td>
<td>85.37 ± 1.34</td>
<td>0.91</td>
</tr>
<tr>
<td>D-KSVD [234]</td>
<td>85.41 ± 1.49</td>
<td>0.92</td>
</tr>
<tr>
<td>FDDL [221]</td>
<td>85.97 ± 1.23</td>
<td>50.03</td>
</tr>
<tr>
<td>DBDL [7]</td>
<td>86.15 ± 1.19</td>
<td>1.20</td>
</tr>
<tr>
<td>JBDC (Proposed)</td>
<td><strong>87.17 ± 0.99</strong></td>
<td>1.18</td>
</tr>
</tbody>
</table>

11.4.2 Object classification

For object classification, we used the Caltech-101 database [74], which contains 9,144 image samples from 101 object categories and a class of background images. The number of samples per class in this database vary between 31 and 800. For classification, we first created 4096-dimensional feature vectors of the images using the 16-layer deep convolutional neural networks for large scale visual recognition [184]. These features were used to create the training and the testing data sets. Following a common evaluation protocol, we used 5, 10, 15, 20, 25 and 30 randomly chosen samples per class for training and the remaining samples were used for testing. Results of our experiments are summarized in Table 11.3. From the table, it is clear that the proposed approach consistently improves the classification accuracy over the existing techniques. The average reduction in the error rate for these experiments is 7.61%. The overall time for classifying 30 samples per class by our approach was 18.77 seconds, whereas DBDL, LC-KSVD and D-KSVD required 18.80, 18.78 and 18.79 seconds, respectively. For JBDC, the final dictionary size was 3001, whereas this value was 3033 for DBDL, and LC-KSVD and D-KSVD required 3030 atoms for the best performance.

11.4.3 Scene categorization

The Fifteen Scene Category database [121] consists of images from fifteen natural scene categories. The average image size in the database is 250 × 300 pixels and the number of sample per class vary between 200 to 400. For this data set, we directly used the 3000-dimensional Spatial Pyramid Features of the images provided by
Table 11.3: Object classification on Caltech-101 [74].

<table>
<thead>
<tr>
<th>Training samples</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRC [213]</td>
<td>76.23</td>
<td>79.99</td>
<td>81.27</td>
<td>83.48</td>
<td>84.00</td>
<td>84.51</td>
</tr>
<tr>
<td>DL-COPAR [203]</td>
<td>76.11</td>
<td>80.40</td>
<td>83.44</td>
<td>84.01</td>
<td>84.85</td>
<td>85.03</td>
</tr>
<tr>
<td>FDDL [221]</td>
<td>78.31</td>
<td>81.37</td>
<td>83.37</td>
<td>84.76</td>
<td>85.66</td>
<td>85.98</td>
</tr>
<tr>
<td>LC-KSVD1 [104]</td>
<td>79.03</td>
<td>82.86</td>
<td>84.13</td>
<td>84.65</td>
<td>86.10</td>
<td>86.94</td>
</tr>
<tr>
<td>D-KSVD [234]</td>
<td>79.69</td>
<td>83.11</td>
<td>84.99</td>
<td>86.01</td>
<td>86.80</td>
<td>87.72</td>
</tr>
<tr>
<td>LC-KSVD [104]</td>
<td>79.74</td>
<td>83.13</td>
<td>85.20</td>
<td>85.98</td>
<td>86.77</td>
<td>87.81</td>
</tr>
<tr>
<td>DBDL [7]</td>
<td>80.11</td>
<td>84.03</td>
<td>85.99</td>
<td>86.71</td>
<td>87.97</td>
<td>88.81</td>
</tr>
<tr>
<td>JBDC (Proposed)</td>
<td><strong>81.64</strong></td>
<td><strong>85.70</strong></td>
<td><strong>86.96</strong></td>
<td><strong>87.88</strong></td>
<td><strong>88.72</strong></td>
<td><strong>89.59</strong></td>
</tr>
</tbody>
</table>

Table 11.4: Classification accuracies (%) on Fifteen Scene Category dataset [121] using Spatial Pyramid Features. The time for computing a single test sample is given in milli-seconds.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy %</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDDL[221]</td>
<td>94.08 ± 0.43</td>
<td>57.99</td>
</tr>
<tr>
<td>D-KSVD [234]</td>
<td>95.12 ± 0.18</td>
<td>0.58</td>
</tr>
<tr>
<td>LC-KSVD1[104]</td>
<td>95.37 ± 0.28</td>
<td>0.59</td>
</tr>
<tr>
<td>SRC [213]</td>
<td>95.41 ± 0.13</td>
<td>78.33</td>
</tr>
<tr>
<td>DL-COPAR [203]</td>
<td>96.02 ± 0.28</td>
<td>55.67</td>
</tr>
<tr>
<td>LC-KSVD[104]</td>
<td>96.38 ± 0.29</td>
<td>0.59</td>
</tr>
<tr>
<td>DBDL[7]</td>
<td>96.98 ± 0.28</td>
<td>0.71</td>
</tr>
<tr>
<td>JBDC (Proposed)</td>
<td><strong>97.73 ± 0.21</strong></td>
<td>0.70</td>
</tr>
</tbody>
</table>

Jiang et al. [104]. From these features, we selected 50 random samples per class for training and used the remaining samples for testing. We summarize the results of our experiments with this data set in Table 11.4. As evident form the table, the proposed approach is also able to improve results for categorizing the natural scenes.

### 11.4.4 Action recognition

We used UCF sports action database [172] for action recognition. The database consists of 10 classes of varied sports actions, having a total of 150 clips @ 10 fps. We used the action bank features [179] for this database to train and test the approaches. Following a common evaluation protocol, we performed a five-fold cross validation. The mean recognition rates of the resulting five experiments are
Table 11.5: Action recognition on UCF Sports Action database [172].

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy %</th>
<th>Method</th>
<th>Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-KSVD [234]</td>
<td>89.1</td>
<td>SRC [213]</td>
<td>92.6</td>
</tr>
<tr>
<td>LC-KSVD1 [104]</td>
<td>89.6</td>
<td>FDDL [221]</td>
<td>93.6</td>
</tr>
<tr>
<td>DL-COPAR [203]</td>
<td>90.7</td>
<td>LDL [220]</td>
<td>95.0</td>
</tr>
<tr>
<td>LC-KSVD [104]</td>
<td>91.7</td>
<td>DBDL [7]</td>
<td>95.1</td>
</tr>
<tr>
<td>JBDC(Proposed)</td>
<td><strong>95.7</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

reported in Table 11.5. For FDDL and DL-COPAR we report the results directly from [220], as our parameter optimization for these algorithms could not achieve these accuracies. Results of LDL [220] are also taken from the original work. The proposed joint Bayesian dictionary and classifier learning approach is able to show an average reduction of 12.2% in the error rate of action recognition.

### 11.5 Discussion

The choice of the parameter values for our approach is intuitive due to its Bayesian nature. In all the experiments, we set $c_o, d_o, e_o$ and $f_0$ to $10^{-6}$. A wide range of similar small values of these parameters (of the non-informative Gamma hyper-priors) results in a very similar performance of the approach. Considering that the data used in our experiments is mainly clean in terms of white noise, we selected $\lambda_{y_o} = \lambda_{h_o} = 10^6$ for the face, object and scene recognition experiments. The values of these precision parameters were set to $10^9$ for the action recognition task due to the less amount of the available training data. Following the common practice in the Beta Process based factor analysis [238], we let $\lambda_{\varphi_o} = 1/L$ and $\lambda_{\psi_o} = 1/C$; and chose $\lambda_{s_o} = \lambda_{t_o} = 1$ for all the experiments. We chose a non-informative initial value for the Bernoulli parameters, i.e. $\pi_{k_o}^c = 0.5, \forall k, \forall c$.

In the light of Lemma 7, we chose $a_o = b_o = \delta/4$ in our experiments, such that $\delta = \min_c |I_c|$. Here, $a_o = b_o$ indicates that we let the final dictionary size to be roughly around the training data size. This rule was empirically derived and it generally worked well for all the recognition tasks in our experiments. The value $\delta/4$ controls the rate at which the dictionary is pruned to its final size - as illustrated in Fig. 11.2 (a). In the figure, we plot the dictionary size obtained after each sampling iteration for a face recognition experiment with Extended YaleB database, where 32 samples per class were used for training. The plot is provided for the first 100 iterations for a better visualization. After around 500 iterations, the recognition...
Figure 11.2: (a) Dictionary size as a function of Gibbs Sampling iterations for Extended YaleB. The first 100 iterations are shown for different values of $a_o$ and $b_o$. (b) Worst-case Point Scale Reduction Factor (PSRF) [75] for $\pi^c_k$, $\forall c$, $\forall k$ as a function of the sampling iterations for Extended YaleB.

rates for all the three curves in Fig. 11.2 (a) were found to be very similar, which is also indicative of good convergence after 500 iterations.

To quantitatively analyze the convergence of the sampling process, we followed Gelman and Rubin [75]. For that, the Potential Scale Reduction Factors (PSRFs) for the key parameters of our model, i.e. $\pi^c_k$, $\forall k$, $\forall c$, were monitored with the increasing number of the sampling iterations, for each recognition task. To compute the PSRF values, we ran 10 sampling processes for each database. Each sampling process was initialized by randomly sampling the parameters $\pi^c_k$ from the standard uniform distribution on the open interval $(0, 1)$. In each experiment, the processes were run for $2n$ iterations and the last $n$ iterations were used to compute the PSRFs. For the details on computing the PSRF values, we refer to [75]. The sampler can be considered converged when PSRF values of the parameters approach to 1. In Fig. 11.2 (b), we show the worst-case values for the Extended YaleB database against the increasing number of the sampler iterations. The worst-case PSFRs are the maximum values among the $C \times |K|$ values for $\pi^c_k$, $\forall k$, $\forall c$. In the figure, these values become very close to 1 after five hundred iterations of the sampler. Since the shown values are for the worst cases, we can conjecture that the performed Gibbs sampler converges reasonably well. The mean PSRFs values for all the five data sets used in our experiments were observed to be very close to 1 after five hundred iterations. Note that, the analysis has been done using those values of the remaining parameters that are mentioned in the preceding paragraphs and using the initialization procedure discussed in Section 11.3.2. The sampling process took around 8 and 23 minutes to
converge for a single experiment of face recognition with Extended YaleB and AR database, respectively. It took around 26, 8 and 3 minutes respectively for a single object, scene and action recognition experiment. For the object recognition, the reported time is for 5 training samples per class.

11.6 Conclusion

We proposed a Bayesian approach to jointly infer a discriminative dictionary and a linear classifier under coupled Beta-Bernoulli processes. Our representation model places separate probability distributions over the dictionary and the classifier, but associates them to the training data using the same Bernoulli distributions. The Bernoulli distributions represent the frequency of the dictionary atom usage in data representation and they are learned adaptively under a Bayesian inference. The inference process also accounts for the class labels and the classifier is tailored according to the learned Bernoulli distributions. The joint inference promotes discriminability in the dictionary, which is further encouraged by using separate Bernoulli distributions to represent the training data of each class in our approach. To classify a test sample, we first compute its representation over the dictionary and then predict its label using the representation with the classifier. The classifier accurately predicts the class label due to its strong coupling with the dictionary. We compared our approach with the state-of-the-art discriminative dictionary learning approaches for face, object, scene and action classification tasks. Experiments demonstrate the effectiveness of the proposed approach across the board.
CHAPTER 12

Non-parametric Coupled Bayesian Dictionary and Classifier Learning for Hyperspectral Classification

Abstract

We present a principled approach to learn a discriminative dictionary along a linear classifier for hyperspectral classification. Our approach places Gaussian Process priors over the dictionary for the relative smoothness of the natural spectra, whereas the classifier parameters are sampled from multi-variate Gaussians. We employ two Beta-Bernoulli processes to jointly infer the dictionary and the classifier. These processes are coupled under the same sets of Bernoulli distributions. In our approach, these distributions signify the frequency of the dictionary atom usage in representing class-specific training spectra, which also makes the dictionary discriminative. Due to the coupling between the dictionary and the classifier, the popularity of the atoms for representing different classes gets encoded into the classifier. This helps in predicting the class labels of test spectra that are first represented over the dictionary by solving a simultaneous sparse optimization problem. The labels of the spectra are predicted by feeding the resulting representations to the classifier. Our approach exploits the non-parametric Bayesian framework to automatically infer the dictionary size - the key parameter in discriminative dictionary learning. Moreover, it also has the desirable property of adaptively learning the association between the dictionary atoms and the class labels by itself. We use Gibbs sampling to infer the posterior probability distributions over the dictionary and the classifier under the proposed model, for which, we derive analytical expressions. To establish the effectiveness of our approach, we test it on benchmark hyperspectral images. The classification performance is compared with the state-of-the-art dictionary learning based classification methods.

Keywords: Hyperspectral classification, coupled Bayesian dictionary learning, discriminative dictionary learning, Gaussian Process, Beta-Bernoulli Process.

*Under review by the IEEE Transactions on Neural Networks and Learning Systems.*
12.1 Introduction

Hyperspectral classification is an active research area in remote sensing [236], [207], [24] due to its applications in mineral identification [34], [142] and aerial surveillance [147], [51]. Researchers have shown that the well-established classification frameworks like logistic regression [124], Support Vector Machines [141], Artificial Neural Networks [16] and k-Nearest Neighbor (k-NN) classifier [131] are able to achieve acceptable performance in hyperspectral classification [190]. Nevertheless, the high dimensionality of hyperspectral signals generally remains a concern for such classical methods.

More recently, the sparse representation framework [155] has attracted significant interest of researchers for classifying the high-dimensional signals [213] - [7]. Wright et al. [213] initially proposed a Sparse Representation based Classification scheme (SRC) that sparsely represents a test signal over an over-complete basis (dictionary) formed by the labelled training data. The label of the test signal is determined to be the class whose associated basis vectors (dictionary atoms) maximally contribute to the signal reconstruction. SRC has shown good performance in face recognition [213], speech recognition [76] as well as in aerial scene classification [52]. Chen et al. [50] also demonstrated successful application of SRC to hyperspectral classification. Recently, Cui and Prasad [58] proposed a class-dependent-SRC for hyperspectral images that combines the concepts of k-NN and SRC.

Whereas the accuracy of SRC is impressive, its computational complexity is very high. This fact has led to numerous efforts in improving the dictionary and the labeling criterion of this scheme. It is now well-established that learning suitable dictionaries, instead of using the training data as the dictionary, can improve both accuracy and efficiency of this scheme [221], [203]. Moreover, the performance can also benefit from learning a classifier along the dictionary instead of using reconstruction fidelity as the labeling criterion [234], [104], [7]. We can broadly categorize the approaches that learn dictionaries for classification (discriminative dictionaries) into three categories.

In the first category, the approaches [222] -[38] make a dictionary discriminative by associating each of its atoms to a single class label. Yang et al. [222] proposed a scheme similar to SRC by learning the dictionary atoms instead of using the training samples as the atoms. Mairal et al. [133] employed the popular K-SVD model [2] in a similar setting and added a discriminative penalty term in the model for the improved classification accuracy. Wang et al. [204] proposed to learn class-specific dictionaries under a similarity constraint over the atoms of the same class and an
incoherence constraint over those that belong to different classes. Similarly, Wu et al. [214] learned active basis model using class-specific training signals and, Castrod and Sapiro [38] associated a dictionary atom to a single class of the training data.

The approaches in the second category [234], [104], [137], [159], [132] allow the dictionary atoms to be shared by different classes. However, the dictionary is encouraged to be discriminative through other means. For instance, Zhang et al. [234] initially proposed to learn a linear classifier along the dictionary. The joint learning process also resulted in making the dictionary discriminative. Similarly, Jiang et al. [104] introduced a ‘discriminative sparse code error’ term in the dictionary learning objective function. Along the joint learning of the classifier with the dictionary, this term further encouraged the dictionary to be discriminative. Mairal et al. [137] and Pham and Venkatesh [159] also proposed to train linear classifiers with the dictionaries to enhance their discriminative abilities. Mairal et al. [132] proposed a Task Driven Dictionary Learning (TDDL) framework to learn discriminative dictionaries for classification. This framework minimizes a joint convex risk function for the representation coefficients, dictionary atoms and the classifier parameters.

The TDDL framework has also been exploited by Sun et al. [190] and Wang et al. [208] for hyperspectral image classification. Sun et al. [190] imposed a structured sparsity prior over the TDDL method for better classification of hyperspectral images. Wang et al. [208] also used the notion of structured sparsity with TDDL in a similar context. Their approach additionally learns a joint classifier with the dictionary. Farani et al. [186] and He et al. [90] learned common dictionaries for all the classes in the training data, but used structured sparsity constraints to benefit from the contextual information in hyperspectral images. All these approaches use neighboring pixels/spectra associated with each training pixel/spectra, where the neighboring pixels are extracted from the test image itself. The contextual information is exploited under the assumption that the test image is available during training and the training spectra are the labeled pixels of the test image.

In the third category of discriminative dictionary learning approaches [203], [240], [181] the dictionaries are formed by concatenating class-specific dictionary atoms and the atoms to commonly represent the data of all the classes. These hybrid approaches are inspired by the extended-SRC model [63] that appends inter-class variation atoms to the dictionary used in SRC. Wang and Kong [203] enforced incoherence between the dictionary atoms learned for different classes and also learned a set of atoms to represent the commonalities of different classes. A Fisher-like regularizer was imposed on the representation coefficients by Zhou and Fan [240] for
learning a hybrid dictionary, whereas a multi-level hybrid dictionary was used by Shen et al. [181] for classification.

In all the aforementioned approaches, the association between the dictionary atoms and the class labels remains the key for effective discriminative dictionaries. Nevertheless, adaptive learning of this association is still under-investigated [220]. Consequently, the learning of a classifier that exploits the adaptively learned associations remains largely unexplored. Moreover, the dictionary sizes in the above mentioned approaches must be fixed beforehand, whereas dependence of dictionary effectiveness on this parameter is observed to be strong [213]. Furthermore, there is no principled way to effectively incorporate hyperspectral domain knowledge into the dictionaries learned by the aforementioned approaches.

In this work, we resolve these problems by exploiting the non-parametric Bayesian framework [157] for learning discriminative dictionaries for hyperspectral data. We propose a Bayesian model to learn the dictionary atoms as samples of Gaussian Processes [167]. Gaussian Processes are particularly suitable for forming hyperspectral dictionaries because they can incorporate the well-known relative smoothness of the spectra with the help of kernels. Our learning model associates the class-labels of the training data to the dictionary atoms using Bernoulli distributions that are adaptively inferred by our approach. These distributions record the frequency of the dictionary atom usage in representing the training data of different classes. Different sets of Bernoulli distributions are used for each class to encourage discrimination in the dictionary. These distributions are also employed in jointly inferring the classifier parameters with the dictionary. Whereas the joint inference also contributes towards making the dictionary discriminative, in our approach, it incorporates the popularity of the dictionary atoms for different classes into the classifier.

Since the popular dictionary atoms for a given class of the training data are likely to be also popular for the test spectra of that class, the classifier can accurately predict the labels of the test sample representations over the learned dictionary. To learn these representations, we solve a simultaneous sparse optimization problem [199] to additionally incorporate the contextual information of the image in the representations. We use Gibbs sampling to perform inference over the proposed Bayesian model. We analytically derive the equations used for the sampling process. Our approach was tested on three popular benchmark data sets for hyperspectral classification and the results are compared with the state-of-the-art dictionary learning/sparse representation based classification approaches. The proposed approach is generally able to out-perform these approaches.
This article is organized as follows. In Section 12.2, we introduce the notations and the relevant concepts from the literature. A detailed account on the proposed approach is given in Section 12.3. In Section 12.4, we present our experiments, and a discussion on parameter values of our approach is provided in Section 12.5. The article concludes in Section 12.6.

12.2 Preliminaries and Background

Let \( x^c_i \in \mathbb{R}^L \) denote the \( i \)th spectra of the \( c \)th class in the training data of \( N \) samples from \( C \) classes. We represent this spectra as: \( x^c_i = \Phi \alpha^c_i + \epsilon_i \), such that \( \Phi \in \mathbb{R}^{L \times |K|} \) is an unknown dictionary, \( \alpha^c_i \in \mathbb{R}^{|K|} \) is the representation of the spectra over the dictionary and \( \epsilon_i \in \mathbb{R}^L \) denotes the noise. We index the \( k \)th atom \( \phi_k \) of the dictionary in a set \( K = \{1,...,K\} \), with unknown cardinality \(|K|\). The training spectra in the \( c \)th class are indexed in set \( I^c \), such that \( \sum_{c=1}^C |I^c| = N \). From the dictionary learning point of view, we are interested in computing a discriminative \( \Phi \), as well as estimating its desired size \(|K|\). Conventional dictionary learning approaches solve the following optimization problem to compute a dictionary:

\[
<\Phi, \alpha_i> = \min_{\Phi, \alpha} ||x_i - \Phi \alpha_i||^2_2 \quad s.t. \quad \forall i, ||\alpha_i||_p \leq t,
\]

where \( ||.||_p \) is the \( \ell_p \)-norm of a vector and \( 't' \) is a constant. Generally, \( p \leq 1 \) to induce sparsity in the representation \( \alpha_i \). Notice that we do not use the superscript \( 'c' \) in (12.1) to indicate that the objective does not discriminate between different classes of the data. We follow the same convention throughout the article and omit \( 'c' \) from a variable if it does not explicitly distinguish between different classes.

It is possible to compute a linear classifier \( \Psi \) that is suitable for classifying the representation \( \hat{\alpha} \) of a test signal over the dictionary that has been computed by (12.1), as follows:

\[
<\Psi> = \min_{\Psi} \sum_{i=1}^N \mathcal{L}\{h_i, f(\alpha_i, \Psi)\} + \lambda ||\Psi||^2_F,
\]

where \( \mathcal{L} \) is the objective loss function, \( \lambda \) denotes the regularizer, \( h_i \in \mathbb{R}^C \) is the label of \( x_i \) and \( f(.) \) is its predicted label. The representation \( \alpha_i \) relates the classifier computed in (12.2) to the dictionary computed in (12.1). While such a classifier can predict a test signal’s label reasonably well by classifying \( \hat{\alpha} \) \[104\], the overall classification strategy is suboptimal as the dictionary completely ignores the class label information of the training data. Hence, Zhang and Li \[234\] and Jiang et al. \[104\] used a concatenation of \( h_i \) and \( x_i \) to learn a basis that serves as a dictionary.
in the first $|\mathcal{K}|$ dimensions and as a classifier in other $C$ dimensions. The use of class label information during dictionary learning resulted in more discriminative bases for their approaches. However, such approaches still need to pre-define the dictionary size, which is a key parameter for effective discriminative dictionaries [213].

A Beta Process [157] can be exploited to remove the necessity of pre-fixing the dictionary size by learning the dictionary non-parametrically. Denoting the base measure of the Beta Process by $h_0$, we can define a finite representation of the process as follows [157]:

$$h = \sum_{k \in \mathcal{K}} \pi_k \delta_{\varphi_k}(\varphi); \quad \pi_k \sim \text{Beta} \left( \pi_k \bigg| \frac{a}{K}, \frac{b(K-1)}{K} \right); \quad \varphi_k \sim h_0,$$

where $a, b$ are the parameters of the process and $\delta_{\varphi_k}(\varphi) = 1$ when $\varphi = \varphi_k$ and 0 otherwise. The finite Beta Process partitions its measurable space into $K$ regions of equal measure. A sample $h$ from the process comprises $|\mathcal{K}|$ probabilities $\pi_k \in \mathcal{K}$ that are associated with the same number of the base measure draws $\varphi_k \in \mathcal{K}$, sampled i.i.d. from $h_0$. If we consider $\pi_k$ to be a Bernoulli distribution parameter, we can use a sample of the Beta Process to further draw a binary vector $z \in \mathbb{R}^{|\mathcal{K}|}$ whose $k$th coefficient follows Bernoulli($\pi_k$). Employing the resulting Beta-Bernoulli Process, we can actually factorize the training samples as: $x_i \approx \Phi z_i, \forall i \in \{1, ..., N\}$ by drawing $N$ binary vectors, such that the atoms of the dictionary $\Phi$ become the base measure draws. Under these settings, the number of the non-zero elements in $z_i$ follow Poisson distribution with the mean parameter value given by the fraction $\frac{a}{b}$, when $K \to \infty$. This automatically decides the desired dictionary size. We refer to [157] for further details in this regard.

Notice that, in a Beta-Bernoulli process the Bernoulli distributions associate the training data with the dictionary atoms using vectors $z_i$. The $k$th Bernoulli distribution (out of $|\mathcal{K}|$) controls the frequency of the $k$th dictionary atom usage in the expansion of the training data. We recently showed [7] that this fact can be exploited to induce discrimination in a Bayesian dictionary by using $|\mathcal{K}|$ distinct Bernoulli distributions in the process for each class separately. Having the same base measure (i.e. dictionary atoms) but $C$ sets of $|\mathcal{K}|$ Bernoulli distribution during dictionary training, naturally results in a discriminative dictionary under a Beta-Bernoulli process. Although we were able to infer a discriminative Bayesian dictionary in [7], the classifier training was performed separately from dictionary learning. Therefore, the inferred dictionary did not fully benefit from the class label information of the training data. Moreover, the base measure used in the model of [7] did not exploit any hyperspectral domain knowledge for the dictionary. In
this work, we overcome these issues with the help of a novel model to jointly learn a non-parametric Bayesian dictionary with a linear classifier. The model employs kernels for the base measure to incorporate the well-known relative smoothness of the natural spectra into the dictionary atoms.

12.3 Proposed Approach

We propose a joint Bayesian dictionary and classifier learning model to infer discriminative dictionaries suitable for hyperspectral data and classifying the representations of natural spectra over the dictionary. The proposed model couples two non-parametric Beta-Bernoulli Processes [157] using common Bernoulli distributions for a joint inference over the dictionary and the classifier. For \( C \) classes, these distributions come in \( C \) distinct sets such that, an element of the \( c^{th} \) set governs the frequency of a dictionary atom usage in the expansion of the training data of the \( c^{th} \) class only. This makes the dictionary learned under the proposed model inherently discriminative. We use Gaussian Processes [167] as the base measure for the dictionary and multivariate Gaussians for the classifier. The Gaussian Processes employ kernels to encourage high correlation among the nearby coefficients of the basis vectors that are used to represent relatively smooth spectral signals in our approach. This results in an effective dictionary for the spectra. On the other hand, the classifier parameters have their own suitable base measure. Moreover, the classifier is also well-tuned to the popularity of the dictionary atoms for representing different class spectra because of the common Bernoulli distributions for the dictionary and the classifier. Since testing spectra also use the popular atoms for their correct class more frequently in their representations over the dictionary, the classifier can accurately predict the label of the test spectra by classifying their representations. We compute these representations by solving a simultaneous sparse optimization problem [199] to also exploit the contextual information in the image.

12.3.1 The Model

Instead of representing a spectra \( x^c_i \) as a binary combination of the basis vectors (as discussed in Section 12.2), we model it as \( x^c_i = \Phi \alpha^c_i + \epsilon_i \), such that \( \alpha^c_i = z^c_i \odot s^c_i \). Here, \( s^c_i \in \mathbb{R}^{K} \) is a weight vector and \( \odot \) denotes an element-wise product. This modeling is possible under a weighted Beta-Bernoulli process [157] and is much more flexible. For each \( x^c_i \), we model its corresponding class label vector as \( h^c_i = \Psi \beta^c_i + h^c_i \epsilon_i \), such that \( \beta^c_i = z^c_i \odot t^c_i \). Here, \( \Psi \) is the classifier, \( h^c_i \epsilon_i \) denotes the modeling error and \( t^c_i \in \mathbb{R}^C \) is a weight vector similar to \( s^c_i \). Notice that, herein, the vector \( z^c_i \) controls
12.3. Proposed Approach

the support (indices of the non-zero elements) of the representations \( \alpha_i^c \) and \( \beta_i^c \), and we have used the same \( z_i^c \) for both of the representations. It means, for a given pair of \( x_i^c \) and \( h_i^c \), only the corresponding basis vectors in \( \Phi \) and \( \Psi \) are activated by the representations. This indicates a coupling between the two bases that we exploit in the following hierarchical Bayesian model for learning the dictionary and the classifier jointly:

\[
\forall i \in \mathcal{I}_c, \forall c:\quad
\begin{align*}
    x_i^c &= \Phi \alpha_i^c + x_i^c \\
    h_i^c &= \Psi \beta_i^c + h_i^c \\
    \alpha_i^c &= z_i^c \odot s_i^c \\
    \beta_i^c &= z_i^c \odot t_i^c
\end{align*}
\]

(12.4)

\[
\begin{align*}
    z_{ik}^c &\sim \text{Bernoulli}(z_{ik}^c | \pi_{ik}^c) \\
    \pi_{ik}^c &\sim \text{Beta}(\pi_{ik}^c | a_o / K, b_o(K - 1) / K) \\
    s_{ik}^c &\sim \mathcal{N}(s_{ik}^c | 0, 1/\lambda_{s_o}) \\
    t_{ik}^c &\sim \mathcal{N}(t_{ik}^c | 0, 1/\lambda_{t_o}) \\
    \varphi_k &\sim \mathcal{GP} (\varphi_k | 0, \Sigma_{k_o}) \\
    \Sigma_{k_o}(\theta_a, \theta_b) &= \frac{1}{\eta_k} \exp \left( -|\theta_a - \theta_b| \right) \\
    \eta_k &\sim \text{Gam}(\eta_k | c_o, d_o) \\
    x_i^c &\sim \mathcal{N}(x_i^c | 0, \Lambda_{x_o}^{-1}) \\
    h_i^c &\sim \mathcal{N}(h_i^c | 0, \Lambda_{h_o}^{-1}).
\end{align*}
\]

For a better readability, we explain the symbols in (12.4) along the discussion on the model below.

In our model, the \( k^{th} \) coefficient \( z_{ik}^c \) of \( z_i^c \) is drawn from a Bernoulli distribution with parameter \( \pi_{ik}^c \), where ‘o’ indicates that the parameter belongs to a prior distribution. We place a conjugate Beta prior over \( \pi_{ik}^c \) to develop a Beta-Bernoulli Process \([157]\), and parameterize the Beta distribution according to (12.3). Our model couples the Beta-Bernoulli Processes for the dictionary and the classifier by using common Beta and Bernoulli distributions. This fact is indicated in (12.3) by aligning the sampling expressions for these distributions to the center. We draw the \( k^{th} \) coefficient \( s_{ik}^c \) of the weight vector \( s_i^c \), from a normal distribution (denoted by \( \mathcal{N} \)), with precision \( \Lambda_{s_o}^{-1} \). Correspondingly, a normal prior, with precision \( \Lambda_{t_o}^{-1} \), is placed over \( t_{ik}^c \). Whereas normal priors in Bayesian models are generally preferred because of their suitable analytical properties, in our experience \([7]\), they work well in practice for a weighted Beta-Bernoulli Process.
In (12.4), we place a zero mean Gaussian Process (GP) prior over the $k^{th}$ dictionary atom $\varphi_k$, that corresponds to the $k^{th}$ base measure draw of the underlying Beta-Bernoulli Process for the dictionary. The matrix $\Sigma_k \in \mathbb{R}^{L \times L}$ denotes the kernel of the GP. Recall, in our settings, the dictionary atoms are the basis vectors for $L$-dimensional spectra. Hence, in the above provided definition of the kernel, $\theta_{\ell}$ signifies the wavelength of the spectra at its $\ell^{th}$ channel. In our model, we use a commonly employed exponential kernel that promotes high correlation between the adjacent spectral channels, with a pre-fixed scaling parameter $\eta_o$. Nevertheless, we introduce a parameter $\eta_k$ in the kernel to also allow it to adjust to the training data. To automatically infer the value of $\eta_k$, we place a non-informative Gamma hyper-prior over this parameter. The Gamma distribution is denoted by Gam in (12.4), that has fixed parameters $\epsilon_o$ and $d_o$.

For the Beta-Bernoulli Process of the classifier, we use multi-variate Gaussian distributions to draw the base measure samples. These prior distributions have zero mean and precision $\Lambda_k \in \mathbb{R}^{C \times C}$. The correspondence between the sampling expressions for the two base measures used in the model is indicated by the braces. Notice the flexibly of the model to allow for the suitable base measures for the subspaces being coupled. Moreover, the model also allows for different noise statistics for the spectra $x_i$ and the label vectors $h_i$. For the former, we sample the error vector from a zero mean Gaussian with precision $\Lambda_x \in \mathbb{R}^{L \times L}$. For the latter, we use $\Lambda_h \in \mathbb{R}^{C \times C}$ as the precision parameter of the Gaussian noise. In (12.4), we use the same noise statistics for the training data sets of all the classes, which is indicated by the omission of the super-script 'c' from $x^*_i$ and $h^*_i$.

For the computational purpose, we further let $\Lambda_x = \lambda_x I_L$, $\Lambda_h = \lambda_h I_C$ and $\Lambda_k = \lambda_k I_C$, where $I_Q$ is an identity matrix in $\mathbb{R}^{Q \times Q}$. This is a common practice for simplifying the Bayesian models and it leads to a considerable computational advantage without significantly degrading the results of our approach. Moreover, following Zhou et al. [238], we also place non-informative Gamma hyper-priors over the Gaussian precision parameters in our model. That is, we let, $\lambda_x, \lambda_h \sim \text{Gam}(\epsilon_o, f_o)$ and $\lambda_k, \lambda_k \sim \text{Gam}(g_o, h_o)$. This makes the inference over the model insensitive to the initialization of the precision parameters. The graphical representation of the complete Bayesian model is given in Fig. 12.1.

### 12.3.2 Inference

We placed suitable prior distributions over the parameters in (12.4) with an aim to infer useful posterior distributions over the dictionary and the classifier. To
compute these posterior distributions, we perform Bayesian inference using Gibbs sampling, that iteratively samples the posterior distributions over the model parameters to arrive at the target distributions. Due to the conjugacy of the probability distributions used in our model, we are able to analytically derive the expressions for sampling the posterior distributions. The derivations of the expressions are provided below in a sequence that is also used to perform the sampling.

**Sampling** $\varphi_k$: According to the proposed model, the posterior distribution over the $k^{th}$ dictionary atom $\varphi_k$ can be given as:

$$p(\varphi_k|\theta) \propto \prod_{i=1}^{N} N(x_{i\varphi_k}|\varphi_k(z_{ik}s_{ik}), \Sigma_{x_i}^{-1}I_L)\mathcal{GP}(\varphi_k|0, \Sigma_{k_o}),$$

where $x_{i\varphi_k} = x_i - \Psi(z_i \odot s_i) + \varphi_k(z_{ik}s_{ik})$ is the contribution of the $k^{th}$ dictionary atom to $x_i$. Using the linear Gaussian model [175], we can show that $\mathcal{GP}(\varphi_k|\mu_k, \Sigma_k)$ should be sampled as the posterior probability distribution over $\varphi_k$ for the Gibbs sampling, where

$$\Sigma_k = \left(\Sigma_{k_o}^{-1} + \lambda_{x_d} \sum_{i=1}^{N} (s_{ik}z_{ik})^2\right)^{-1}, \quad \mu_k = \lambda_{x_d} \Sigma_k \sum_{i=1}^{N} (s_{ik}z_{ik})x_{i\varphi_k}.$$ 

In the above expressions, we omit the super-script ‘c’ because the dictionary atoms are sampled based on the training data from all the classes simultaneously. This is also true for sampling $\eta_k$ and the classifier parameters, as discussed below.
Chapter 12. Bayesian Dictionary and Classifier Learning for HS Classification

Sampling \( \eta_k \): According to (12.4), the posterior distribution over \( \eta_k \) is proportional to \( G\mathcal{P} \left( \Phi | 0, \frac{1}{\eta_k} \Sigma_k \right) \text{Gam}(\eta_k | a_o, d_o) \), where \( \Sigma_k = \exp(-|\theta_b - \theta_o|/\eta_o) \). This multiplication is proportional to \( \frac{\nu_n}{\Gamma(a_n)} \left( \frac{1}{\eta_k} \right)^{a_n} \exp \left\{ -\frac{n}{\eta_k} \varphi_k^T \Sigma_k \varphi_k - b_o \eta_k \right\} \). Hence, we use the following Gamma distribution to sample \( \eta_k \): \( \text{Gam} \left( \eta_k | a_o + \frac{n}{2}, b_o + \frac{1}{2} \varphi_k^T \Sigma_k \varphi_k \right) \).

Sampling \( \psi_k \): Using similar reasoning as for the dictionary atoms, we can sample the \( k^{th} \) column \( \psi_k \) of the classifier parameters from \( \mathcal{N}(\psi_k | \mu_k, \lambda_k^{-1} I_C) \), such that:

\[
\lambda_k = \lambda_k + \lambda_{ho} \sum_{i=1}^N (z_{ik} t_{ik})^2, \quad \mu_k = \lambda_{ho} \lambda_k^{-1} \sum_{i=1}^N (z_{ik} t_{ik}) h_{i0}^T.
\]

Sampling \( z_{ik} \): According to the proposed model, the posterior probability distribution over \( z_{ik} \) can be given as:

\[
p(z_{ik} = 1) \propto \mathcal{N}(x_{i0}^c | \varphi_k(z_{ik}^c s_{ik}^c), \lambda_{x0}^{-1} I_L) \mathcal{N}(h_{i0}^c | \psi_k(z_{ik}^c t_{ik}^c), \lambda_{ho}^{-1} I_C) \text{Bernoulli}(z_{ik}^c | \eta_{ko}).
\]

Based on this expression, we can show that \( z_{ik} \) should be sampled for inference from a normalized Bernoulli distribution as follows:

\[
z_{ik} \sim \text{Bernoulli} \left( \frac{\pi_{ko}^c \xi_1 \xi_2}{1 - \pi_{ko}^c + \xi_1 \xi_2 \pi_{ko}^c} \right), \quad \xi_1 = \exp \left( -\frac{\lambda_{ho}^c}{2} (\varphi_k^T \varphi_k s_{ik}^c z_{ik}^c - 2 s_{ik}^c x_{i0}^c) \right), \quad \xi_2 = \exp \left( -\frac{\lambda_{ho}^c}{2} (\psi_k^T \psi_k t_{ik}^c z_{ik}^c - 2 t_{ik}^c h_{i0}^c \varphi_k) \right).
\]

Sampling \( s_{ik} \): The following can be written regarding the posterior probability distribution over \( s_{ik}^c \):

\[
p(s_{ik}^c | \cdot) \propto \mathcal{N}(x_{i0}^c | \varphi_k(z_{ik}^c s_{ik}^c), 1/\lambda_{x0}^c I_L) \mathcal{N}(s_{ik}^c | 0, 1/\lambda_{ho}^c).
\]

Exploiting the conjugacy between the distributions, \( s_{ik}^c \) can be sampled from the normal distribution \( \mathcal{N}(s_{ik}^c | \mu_{s}^c, 1/\lambda_{s}^c) \), where:

\[
\lambda_s^c = \lambda_{ho}^c + \lambda_{x0}^c z_{ik}^c \varphi_k^T \varphi_k, \quad \mu_s^c = (1/\lambda_s^c) \lambda_{x0}^c z_{ik}^c \varphi_k^T x_{i0}^c.
\]

Sampling \( t_{ik} \): Similarly, we can sample the weight \( t_{ik}^c \) from \( \mathcal{N}(t_{ik}^c | \mu_t^c, 1/\lambda_t^c) \), where:

\[
\lambda_t^c = \lambda_{ho}^c + \lambda_{x0}^c z_{ik}^c \psi_k^T \psi_k, \quad \mu_t^c = (1/\lambda_t^c) \lambda_{ho}^c z_{ik}^c \psi_k^T h_{i0}^c.
\]

Sampling \( \pi_k^c \): The posterior probability over \( \pi_k^c \) can be written as follows:

\[
p(\pi_k^c | \cdot) \propto \prod_{i \in I_C} \text{Bernoulli}(z_{ik}^c | \pi_k^c) \text{Beta} \left( \pi_k^c | a_o, b_o(K - 1)/K \right).
\]
Using the conjugacy between the Beta and the Bernoulli distributions, we sample $\pi_c^k$ from the following Beta distribution:

$$\text{Beta} \left( \frac{a_o}{K} + \sum_{i=1}^{\left| I_c \right|} z_{ik}^c, \frac{b_o(K - 1)}{K} + \left| I_c \right| - \sum_{i=1}^{\left| I_c \right|} z_{ik}^c \right).$$

**Sampling $\lambda_c^s$:** In order to sample $\lambda_c^s$, we treat $s_{ik}^c$ for all the dictionary atoms simultaneously (the same is done for $\lambda_c^t$ below). We assume the vector $s_i^c$ to be drawn from a Gaussian distribution with isotropic precision. This simplification allows us to efficiently sample the posterior distribution over $\lambda_c^s$, which is given as:

$$p(\lambda_c^s | \cdot) \propto \prod_{i \in I_c} \mathcal{N}(s_i^c | 0, 1/\lambda_c^s, I_{|K|}) \text{Gam}(\lambda_c^s | g_o, h_o).$$

Exploiting the conjugacy between the Gaussian and the Gamma distributions, we sample $\lambda_c^s$ as follows:

$$\lambda_c^s \sim \text{Gam} \left( \frac{\left| I_c \right| |K|}{2} + g_o, \frac{1}{2} \sum_{i=1}^{\left| I_c \right|} ||s_i^c||^2_2 + h_o \right).$$

**Sampling $\lambda_c^t$:** Correspondingly, we can sample $\lambda_c^t$ from the Gamma probability distribution mentioned in the above expression, with $t_i^c$ replacing $s_i^c$.

**Sampling $\lambda_x$:** The posterior probability distribution over $\lambda_x$ can be written as:

$$p(\lambda_x | \cdot) \propto \prod_{i=1}^{N} \mathcal{N}(x_i | \Phi(z_i \odot s_i), \lambda_{x_o}^{-1} I_L) \text{Gam}(\lambda_x | e_o, f_o).$$

Similar to the case of $\lambda_c^s$, we can show that $\lambda_x$ must be sampled from the following Gamma distribution:

$$\text{Gam} \left( \frac{LN}{2} + e_o, \frac{1}{2} \sum_{i=1}^{N} ||x_i - \Phi(z_i \odot s_i)||_2^2 + f_o \right).$$

**Sampling $\lambda_h$:** Analogously, we sample $\lambda_h$ from the following Gamma distribution:

$$\text{Gam} \left( \frac{CN}{2} + e_o, \frac{1}{2} \sum_{i=1}^{N} ||h_i - \Psi(z_i \odot t_i)||_2^2 + f_o \right).$$

We initialize the Gibbs sampling process by randomly choosing the training spectra with replacement as the dictionary atoms. We represent the training data over the initial dictionary using the Orthogonal Matching Pursuit algorithm (OMP) [33]
and use them as the initial $s_i^c$ and $t_i^c$. The initial values for $z_i^c$ are obtained by replacing the non-zero coefficients of the sparse codes with ones. The initial classifier parameters are computed by solving (12.2) using the same codes. This initialization procedure is inspired by the existing discriminative dictionary learning approaches [104], [234], [7].

12.3.3 Dictionary Size

As a result of the Bayesian inference, we determine the posterior distributions over the dictionary atoms in the form of Gaussian Processes. We use the mean parameters of these distributions as the desired dictionary. The classifier is also formed by the mean parameters of the posterior distributions over $\psi_k$, $\forall k$. In regards of the desired dictionary/classifier size, let us consider the covariances of $x_i^c$ and $h_i^c$ under the proposed representation model. These covariances are given as $E[x_i^c x_i^c\top] = \frac{a_o K}{a_o + b_o (K - 1)} \Sigma_{x_o} + \frac{I_C}{\lambda_{x_o}}$ and $E[h_i^c h_i^c\top] = \frac{a_o K}{a_o + b_o (K - 1)} \Lambda_{h_o}^{-1} + \frac{I_C}{\lambda_{h_o}}$, respectively. Recall that, in a Beta Process, $K$ signifies the number of partitions of the measurable space. In our settings, these partitions correspond to the dictionary atoms (and vectors $\psi_k$). Letting $K \to \infty$ results in $E[x_i^c x_i^c\top] \to \frac{a_o}{b_o} \Sigma_{x_o} + \frac{I_C}{\lambda_{x_o}}$ and $E[h_i^c h_i^c\top] \to \frac{a_o \Lambda_{h_o}^{-1}}{b_o} + \frac{I_C}{\lambda_{h_o}}$. Firstly, this shows that our joint model remains well defined in the infinite limit. Moreover, it indicates that, for our model, the expected number of the partitions utilized by the underlying Beta Processes to represent a training data pair $x_i^c, h_i^c$ are given by $\frac{a_o}{b_o}$. We arrive at this conclusion by ignoring the fraction $F = \frac{a_o}{a_o + b_o (K - 1)}$ in the expressions for the covariances of $x_i^c$ and $h_i^c$, and comparing the resulting expressions with those obtained by letting $K \to \infty$. We ignore $F$ because it only appears due to $z_i^c$ in (12.4). This line of reasoning and the result are in accordance with those presented by Paisley and Carin [157] in the original proposal of the Beta Process. However, we deal with a joint model and class-specific training data.

The above discussion indicates that a finite dictionary and classifier size is expected for our model to represent a finite amount of training data. We present Lemma 8 regarding the Gibbs sampling process in Section 12.3.2 to automatically determine this size.

Lemma 8. When $\sum_{c=1}^{C} \pi_k^c \to 0$ in a given iteration of the Gibbs sampling process, $E[\pi_k^c] \to 0$ in the later iterations of the process, when $0 < a_o, b_o \leq |I_c| \ll K$.

Proof: When $\sum_{c=1}^{C} \pi_k^c \to 0$, $\sum_{i=1}^{|I_c|} z_{ik}^c \to 0, \forall c$ in the next iteration of the sampling process. This results in the posterior distribution over $\pi_k^c$ to be approximated by
12.4. Experiments

\[
\beta \left( \pi^c_k \|a_o/K, b_o(K-1)/K + |I_c| \right), \text{ for which, } E[\pi^c_k] = \frac{a_o + b_o(K-1)/K |I_c|}{a_o + b_o(K-1)/K + |I_c|}. \text{ For } 0 < a_o, b_o \leq |I_c| \ll K, E[\pi^c_k] \to 0.
\]

Considering Lemma 8, we let \( K = 1.25 \times N \) to initialize the sampling process and drop the \( k \)th dictionary atom (and \( \psi_k \)) before the next sampling iterations if \( \sum_{c=1}^{C} \pi^c_k < 10^{-6} \) in the current iteration. By gradually dropping the redundant basis vectors, the dictionary and the classifier automatically adjust to their desired size to represent the data.

12.3.4 Classification

To predict the label of a test pixel/spectra, we first extract a \( p \times p \) patch (along the spatial dimensions) from the hyperspectral image, with the test pixel at the center. Then, we compute the sparse codes \( \hat{\alpha}_j \in \{1, \ldots, p^2\} \) of these pixels over the learned dictionary. Afterwards, we compute a vector \( \ell \in \mathbb{R}^C \) as follows:

\[
\ell = \sum_{j=1}^{p^2} \Psi \hat{\alpha}_j, \quad (12.5)
\]

and decide the label of the pixel by maximizing the coefficients of the vector \( \ell \).

In the above mentioned strategy, we use a patch from the image to exploit the contextual information in the scene. Our assumption is that most of the pixels around the test pixel are composed of the same material as the test pixel. This means, we expect most of the pixels in the patch to activate the same/similar dictionary atoms in their representations \( \hat{\alpha}_j, \forall j \). Therefore, we compute these vectors by solving the following simultaneous sparse optimization problem, using the Simultaneous-OMP algorithm (SOMP) [199]:

\[
< \hat{\mathbf{A}} > = \min_{\hat{\mathbf{A}}} ||\hat{\mathbf{X}} - \Phi \hat{\mathbf{A}}||_F \quad \text{ s.t. } ||\hat{\mathbf{A}}||_{row, 0} \leq K_o, \quad (12.6)
\]

where \( \hat{\mathbf{A}} = [\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_{p^2}] \in \mathbb{R}^{[K]\times p^2}, \hat{\mathbf{X}} \in \mathbb{R}^{L\times p^2} \) contains the pixels in the patch, \( ||.||_{row, 0} \) counts the non-zero rows in a matrix and \( K_o \) is a pre-defined sparsity threshold. Solving (12.6) ensures that similar dictionary atoms are used by the representations of all the pixels in the patch, which encourages the popular dictionary atoms for the majority class in the patch to be more active in the representations. Since \( \Psi \) is already encoded with the activity level of the atoms for different classes, it accurately predicts the label of the test spectra under the proposed strategy.

12.4 Experiments

We evaluated our approach on three benchmark data sets for hyperspectral image classification. The images in these data sets belong to Indian Pines scene, Salinas
scene and the Pavia University scene, see Fig. 12.2. The images are available for
Remote_Sensing_Scenes. We compare the results of our approach with the state-
of-the-art dictionary learning/sparse representation based classification methods.
These approaches include the Discriminative K-SVD algorithm (DKSVD) [234], the
Label-Consistent K-SVD algorithm (LC-KSVD) [104], the Sparse Representation
based Classification scheme (SRC) [213] and the Discriminative Bayesian Dictio-
nary Learning approach (DBDL) [7]. Similar to our approach, D-KSVD, LC-KSVD
and DBDL learn classifiers along the dictionaries. Therefore, we can also enhance
these approaches by replacing their criteria for predicting the label of a test spectra
by the strategy we proposed for our approach in Section 12.3.4. We include the re-
sults of these enhancements as well. Note that, the enhancements over the original
methods are contribution of this work. We indicate this fact below by annotating
the enhanced approaches with the symbol ‡.

We used the public code of LC-KSVD [104] provided by the original authors. We
modified this code to implement D-KSVD, as suggested by Jiang et al. [104]. We
also used the original code of DBDL in our experiments. The SPAMS toolbox [135]
was used for implementing SRC. The same toolbox was also used to solve (12.6)
for our approach and for the enhancements of the existing approaches. We used
the OMP implementation provided by Elad et al. [177] for the sparse coding in LC-
KSVD, D-KSVD and DBDL. The parameters of all these approaches were carefully
optimized in our experiments using the cross-validation method. For the LC-KSVD
and D-KSVD, we fixed the dictionary sizes to 80% of the training data size, as
this generally resulted in the best accuracies for these approaches. We also fixed
the sparsity threshold of SOMP and OMP for all the approaches, including the
proposed approach, to 25. Generally, the results did not vary significantly for the sparsity thresholds above 20 for any approach. However, 25 was chosen to ensure the best performance. We defer the discussion on the parameter value selection of the proposed approach to Section 12.5. To evaluate the classification performance, we employ three commonly used metrics. Namely, the overall classification accuracy (OA), the average of the classification accuracies for each class (AA) and the $\kappa$ coefficient, that computes the fraction of the correctly classified test pixels, corrected by the number of the agreements that would be expected purely by chance. The definition of $\kappa$ is given below.

$$\kappa = \frac{OA - R}{1 - R},$$

(12.7)

where $R = \sum_{c=1}^{C} A^c B^c$, such that $A^c$ is the ratio of the test samples assigned to the $c^{th}$ class by the algorithm and $B^c$ denotes the ratio of the test samples that actually belong to that class. Higher $\kappa$ values indicate better classification.

12.4.1 Indian Pines Image

The $145 \times 145 \times 224$ Indian Pines image is generated by the NASA’s Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) [80]. A pixel of this image is a spectra with 224 bands in the wavelength range 0.2 to 2.4 µm. Following the common practice, 20 water absorption bands were removed from the image before we performed the experiments. There are 16 classes in the image for whom the label is provided. The first column of Table 12.1 indicates these labels. In our experiments, we randomly chose $\sim 10\%$ of the labeled pixels as the training data, ensuring that the selection is made from every class. The remaining $\sim 90\%$ pixels were used as the test spectra. The details on the training/testing data distributions for each class are given in the second column of Table 12.1. The table also summarizes the results of our experiments. The remaining columns show the classification accuracies in percentages for each approach. The best results are bold-faced. From these results, it is clear that the proposed approach generally out-performed the existing techniques and their enhancements.

The overall second best performance is shown by DBDL [7] when it is enhanced with our label prediction strategy, presented in Section 12.3.4. However, our approach still resulted in a 23.7% reduction in the overall error rate of DBDL‡. In our opinion, this reduction is primarily the result of using more appropriate base measure for the dictionary and a better use of the class label information in the
Table 12.1: Classification accuracy (%) for the Indian Pines Image

<table>
<thead>
<tr>
<th>Class</th>
<th>Training/testing</th>
<th>SRC</th>
<th>D-KSVD</th>
<th>LC-KSVD</th>
<th>DBDL</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfalfa</td>
<td>10/36</td>
<td>33.92</td>
<td>35.29</td>
<td>100.0</td>
<td>35.00</td>
<td>100.0</td>
</tr>
<tr>
<td>Corn-no till</td>
<td>143/1285</td>
<td>51.14</td>
<td>47.94</td>
<td>85.77</td>
<td>49.46</td>
<td>87.43</td>
</tr>
<tr>
<td>Corn-min till</td>
<td>83/747</td>
<td>51.43</td>
<td>49.67</td>
<td>83.85</td>
<td>49.25</td>
<td>88.10</td>
</tr>
<tr>
<td>Corn</td>
<td>24/213</td>
<td>42.63</td>
<td>34.69</td>
<td>86.88</td>
<td>42.20</td>
<td>79.18</td>
</tr>
<tr>
<td>Grass/Pasture</td>
<td>48/435</td>
<td>70.16</td>
<td>75.11</td>
<td>95.59</td>
<td>42.63</td>
<td>89.47</td>
</tr>
<tr>
<td>Grass/Trees</td>
<td>73/657</td>
<td>85.50</td>
<td>83.21</td>
<td>90.07</td>
<td>49.25</td>
<td>85.46</td>
</tr>
<tr>
<td>Grass/Pasture-mowed</td>
<td>10/18</td>
<td>100.0</td>
<td>94.35</td>
<td>100.0</td>
<td>93.18</td>
<td>94.95</td>
</tr>
<tr>
<td>Hay-windrowed</td>
<td>49/480</td>
<td>100.0</td>
<td>94.35</td>
<td>100.0</td>
<td>93.18</td>
<td>94.95</td>
</tr>
<tr>
<td>Hayed</td>
<td>38/70</td>
<td>85.50</td>
<td>83.21</td>
<td>90.07</td>
<td>49.25</td>
<td>85.46</td>
</tr>
<tr>
<td>Hayed/no till</td>
<td>77/657</td>
<td>70.16</td>
<td>75.11</td>
<td>95.59</td>
<td>42.63</td>
<td>89.47</td>
</tr>
<tr>
<td>Hayed/no till</td>
<td>24/213</td>
<td>42.63</td>
<td>34.69</td>
<td>86.88</td>
<td>42.20</td>
<td>79.18</td>
</tr>
<tr>
<td>Hayed/no till</td>
<td>100.0</td>
<td>38.46</td>
<td>35.00</td>
<td>100.0</td>
<td>35.00</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Avg. Acc. (AA) % | 60.76 | 58.54 | 61.77 | 60.35 | 59.91 | 59.17 |

Overall Acc. (OA) % | 65.17 | 63.80 | 63.71 | 63.71 | 63.57 | 63.07 |

κ | 0.621 | 0.608 | 0.607 | 0.607 | 0.612 | 0.591 |

Chapter 12. Bayesian Dictionary and Classifier Learning for HS Classification
12.4. Experiments

Figure 12.3: Comparison of the dictionary atoms learned by the proposed approach and DBDL [7]. The atom learned by our approach with a Gaussian Process prior is more similar to the ground truth due to its relative smoothness.

Joint learning process. In Fig. 12.3, we exemplify the appropriateness of Gaussian Processes as the base measure for a hyperspectral dictionary. The figure plots representative dictionary atoms in the wavelength range 1.55 to 2.4 µm learned by DBDL and the proposed approach. The atoms correspond to the material ‘Alfafa’, for which the reflectance spectra is also shown. Notice the relative smoothness of the spectra that is more appropriately captured by the proposed approach as compared to DBDL. In the figure, we selected a partial wavelength range for better visibility of the differences between the spectra.

One interesting aspect of Table 12.1 is the performance gain achieved by using the classification strategy proposed in Section 12.3.4 instead of using the original labeling criterion of the existing approaches. The significant gain is the result of exploiting the spatial information in the image by using the patch size $5 \times 5$ in our classification strategy. Well-defined spatial patterns in the ground truth for the test image are visible in Fig. 12.4, where we also show the classification maps of the four approaches that consider the spatial information. By taking advantage of the well-structured spatial patterns, the labels of the test pixels are predicted with much higher accuracy by the approaches.

We note that techniques like [208] - [90] have reported better results on the Indian Pines image with roughly similar testing and training data distributions. However, these approaches also exploit the spatial patterns of the test image during the dictionary training phase. They make a strong assumption that the test image is also available during dictionary training and the training spectra are always the labeled pixels of the test image. Therefore, for each training pixel, they extract a patch of the neighboring pixels from the test image and assume that the complete
Figure 12.4: Classification maps for the Indian Pines image: The training and testing (ground truth) pixels and their labels are shown, along the label maps predicted by the proposed approach, DBDL [7], LC-KSVD [104] and D-KSVD [234]. For the existing approaches, we replace the label prediction criterion with our classification strategy (indicated by ‡). The label numbers are in accordance with Table 12.1. The unlabeled pixels are shown as zeros.

In this work, we do not allow augmentation of the training data with the test data before dictionary learning. Moreover, we consider training pixels to be individual spectra that do not necessarily spatially correlate with each other or the test data. Our comparison includes only those approaches that follow the machine learning convention of treating the training and the testing data separately.

12.4.2 Salinas Image

This image is also collected by the AVIRIS over Salinas, CA. The original image size is $512 \times 217 \times 224$. Out of the 224 spectral channels, we drop 20 water ab-

\footnote{For the considered image, the test samples may have some spatial correlation with the training data. However, this correlation is not exploited by our approach, or any other approach considered in the comparison.}
Table 12.2: Classification accuracy (%) for the Salinas Image

<table>
<thead>
<tr>
<th>Class</th>
<th>Training/testing</th>
<th>SRC</th>
<th>D-KSVD</th>
<th>D-KSVD</th>
<th>LC-KSVD</th>
<th>LC-KSVD</th>
<th>DBDL</th>
<th>DBDL</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1- Weeds 1</td>
<td>20/1989</td>
<td>97.11</td>
<td>96.82</td>
<td>99.94</td>
<td>96.30</td>
<td>99.84</td>
<td>98.99</td>
<td>99.84</td>
<td>100.0</td>
</tr>
<tr>
<td>2- Weeds 2</td>
<td>37/3689</td>
<td>98.49</td>
<td>97.98</td>
<td>99.24</td>
<td>98.11</td>
<td>99.56</td>
<td>99.28</td>
<td>99.70</td>
<td>99.75</td>
</tr>
<tr>
<td>3- Fallow</td>
<td>20/1956</td>
<td>89.19</td>
<td>87.02</td>
<td>97.54</td>
<td>89.45</td>
<td>97.27</td>
<td>83.43</td>
<td>99.67</td>
<td>99.40</td>
</tr>
<tr>
<td>4- Fallow plow</td>
<td>14/1380</td>
<td>84.00</td>
<td>80.28</td>
<td>96.34</td>
<td>84.08</td>
<td>96.46</td>
<td>79.34</td>
<td>98.72</td>
<td>98.77</td>
</tr>
<tr>
<td>5- Fallow smooth</td>
<td>27/2651</td>
<td>88.72</td>
<td>87.29</td>
<td>95.05</td>
<td>88.93</td>
<td><strong>95.38</strong></td>
<td>83.05</td>
<td>94.70</td>
<td>95.08</td>
</tr>
<tr>
<td>6- Stubble</td>
<td>40/3919</td>
<td>99.71</td>
<td>99.54</td>
<td>99.79</td>
<td>99.74</td>
<td>99.77</td>
<td>99.64</td>
<td>99.97</td>
<td><strong>100.0</strong></td>
</tr>
<tr>
<td>7- Celery</td>
<td>36/3543</td>
<td>98.14</td>
<td>98.16</td>
<td>99.80</td>
<td>97.84</td>
<td>99.91</td>
<td>99.71</td>
<td><strong>99.97</strong></td>
<td>99.88</td>
</tr>
<tr>
<td>8- Grapes</td>
<td>113/11158</td>
<td>66.62</td>
<td>65.36</td>
<td>74.14</td>
<td>65.78</td>
<td>73.36</td>
<td>67.44</td>
<td>78.02</td>
<td><strong>78.10</strong></td>
</tr>
<tr>
<td>9- Soil</td>
<td>62/6141</td>
<td>87.31</td>
<td>87.13</td>
<td>97.64</td>
<td>88.23</td>
<td>97.64</td>
<td>84.51</td>
<td>98.14</td>
<td><strong>98.51</strong></td>
</tr>
<tr>
<td>10- Corn</td>
<td>33/3245</td>
<td>82.59</td>
<td>81.79</td>
<td>96.32</td>
<td>84.71</td>
<td>96.80</td>
<td>83.16</td>
<td>96.66</td>
<td><strong>97.22</strong></td>
</tr>
<tr>
<td>11- Lettuce 4wk</td>
<td>11/1057</td>
<td>96.48</td>
<td>98.66</td>
<td><strong>99.90</strong></td>
<td>97.55</td>
<td>99.70</td>
<td>92.99</td>
<td>99.61</td>
<td>99.52</td>
</tr>
<tr>
<td>12- Lettuce 5wk</td>
<td>19/1908</td>
<td>83.54</td>
<td>78.75</td>
<td>97.72</td>
<td>83.13</td>
<td>98.58</td>
<td>77.82</td>
<td>97.98</td>
<td><strong>98.69</strong></td>
</tr>
<tr>
<td>13- Lettuce 6wk</td>
<td>9/907</td>
<td>69.97</td>
<td>68.30</td>
<td>98.24</td>
<td>68.79</td>
<td>96.63</td>
<td>65.44</td>
<td>98.63</td>
<td><strong>99.08</strong></td>
</tr>
<tr>
<td>14- Lettuce 7wk</td>
<td>11/1059</td>
<td>78.38</td>
<td>71.10</td>
<td>93.34</td>
<td>72.90</td>
<td>94.53</td>
<td>76.92</td>
<td>95.08</td>
<td><strong>95.19</strong></td>
</tr>
<tr>
<td>15- Vinyard untrained</td>
<td>73/7195</td>
<td>57.44</td>
<td>56.57</td>
<td>84.08</td>
<td>56.10</td>
<td><strong>84.50</strong></td>
<td>57.03</td>
<td>84.23</td>
<td>83.94</td>
</tr>
<tr>
<td>16- Vineyard trellis</td>
<td>18/1789</td>
<td>99.88</td>
<td>99.59</td>
<td><strong>100.0</strong></td>
<td>99.48</td>
<td><strong>100.0</strong></td>
<td>99.77</td>
<td>100.0</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Average Accuracy (AA) % 86.09 84.65 95.58 85.69 95.63 84.28 95.89 **96.45**
Overall Accuracy (OA) % 81.87 80.90 90.86 81.63 90.74 81.15 91.98 **92.75**

κ 0.803 0.792 0.899 0.800 0.898 0.795 0.909 **0.918**
Chapter 12. Bayesian Dictionary and Classifier Learning for HS Classification

Figure 12.5: Classification maps for the Salinas image: The training pixels and the ground truth are shown along the label maps predicted by our approach, DBDL [7], LC-KSVD [104] and D-KSVD [234]. We replaced the labeling criterion of the existing approaches with our label prediction strategy to achieve these results (indicated by ‡). The class label numbers are in accordance with Table 12.2. The unlabeled pixels are assigned label 0.

Absorption channels for this image as well. The main contents of the imaged scene are vegetables, soils and vineyard fields. The ground truth for this image has 16 labels, given in the first column of Table 12.2. The second column of the table shows the used training and testing pixel distributions for the classes. In our experiments, we used randomly chosen \( \sim 1\% \) of the data for training and the remaining \( \sim 99\% \) for testing. We used the patch size 3 \( \times \) 3 for all the approaches that exploit the spatial information in the scene. Larger patch sizes did not affect the results significantly.

The results of our experiments are also summarized in Table 12.2. Similar to the Indian Pines image, our approach generally performed better than the existing approaches for this image as well. The best accuracies for most of the individual classes were obtained by the proposed approach. Over DBDL [7], that showed the second best results, the proposed approach reduced the error in the Average Accuracy by 13.6\% and in Overall Accuracy by 9.6\%. We illustrate the classification maps obtained by the proposed approach, along the enhancements of the other approaches in Fig. 12.5.

12.4.3 Pavia University Image

This image is taken by the Reflective Optics System Imaging Spectrometer [http://www.a-a-r-s.org/aars/proceeding/ACRS1989/Papers/ENV89-2.htm] over the Pavia University, northern Italy. It has 115 spectral channels in the wavelength range 0.43 to 0.86 \( \mu m \), out of which 12 noisy channels have been dropped in our experiments, following the common protocol. For the used 610 \( \times \) 340 \( \times \) 103 image, the
### Table 12.3: Classification accuracy (%) for the Pavia University Image

<table>
<thead>
<tr>
<th>Class</th>
<th>SRC</th>
<th>D-KSVD</th>
<th>LC-KSVD</th>
<th>DBDL</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1- Asphalt</td>
<td>548/6304</td>
<td>94.87</td>
<td>94.71</td>
<td>96.45</td>
<td>96.53</td>
</tr>
<tr>
<td>2- Meadows</td>
<td>540/18146</td>
<td>92.17</td>
<td>91.95</td>
<td>97.38</td>
<td>87.87</td>
</tr>
<tr>
<td>3- Gravel</td>
<td>392/1815</td>
<td>58.12</td>
<td>58.46</td>
<td>74.24</td>
<td>78.79</td>
</tr>
<tr>
<td>4- Trees</td>
<td>524/2912</td>
<td>80.11</td>
<td>77.51</td>
<td>93.73</td>
<td>94.46</td>
</tr>
<tr>
<td>5- Metal sheets</td>
<td>265/1113</td>
<td>99.17</td>
<td>98.62</td>
<td>97.91</td>
<td>99.26</td>
</tr>
<tr>
<td>6- Bare soil</td>
<td>552/4572</td>
<td>56.23</td>
<td>53.87</td>
<td>77.50</td>
<td>82.30</td>
</tr>
<tr>
<td>7- Bitumen</td>
<td>375/2364</td>
<td>80.27</td>
<td>77.83</td>
<td>90.01</td>
<td>95.02</td>
</tr>
<tr>
<td>8- Bricks</td>
<td>514/3264</td>
<td>99.84</td>
<td>99.84</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>9- Shadows</td>
<td>231/795</td>
<td>99.84</td>
<td>99.84</td>
<td>100.00</td>
<td>100.00</td>
</tr>
</tbody>
</table>

**Average Accuracy (AA) %**
- SRC: 96.15
- D-KSVD: 93.77
- LC-KSVD: 92.78
- DBDL: 92.88
- Proposed: 96.53

**Overall Accuracy (OA) %**
- SRC: 96.15
- D-KSVD: 93.77
- LC-KSVD: 92.78
- DBDL: 92.88
- Proposed: 96.53

**κ**
- SRC: 0.747
- D-KSVD: 0.735
- LC-KSVD: 0.874
- DBDL: 0.743
- Proposed: 0.736

**κ**
- SRC: 0.747
- D-KSVD: 0.735
- LC-KSVD: 0.874
- DBDL: 0.743
- Proposed: 0.736
Figure 12.6: Classification maps for the Pavia University image: The training pixels, ground truth and the classification map of the proposed approach are given. The class label numbers are in accordance with Table 12.3. ‘0’ indicates unlabeled pixels.

ground truth contains nine classes that are given in the first column of Table 12.3. In our experiments, we randomly selected $\sim 10\%$ of the labeled pixels for training and the remaining were used for testing. The second column of Table 12.3 gives details on the training and testing distributions for each class. The remaining columns of the table show the classification accuracies (%) for each technique in the comparison. As can be observed, the results on this image also favor the proposed approach. For classification, we used a patch size of $3 \times 3$ for our approach and the approaches annotated with $\dagger$. In Fig. 12.6, we show the classification map predicted by our approach along the ground truth and the training pixel map.

12.5 Discussion

Due to the Bayesian nature of our model, the selection of the parameter values for our approach is intuitive. For the non-informative Gamma distributions, we selected the value $10^{-6}$ for the parameters $c_o, d_o, e_o, f_o, g_o$ and $h_o$ in all the experiments. The approach is insensitive to the small perturbations in the values of these parameters, which can be easily verified by analyzing the expressions of the posterior distributions for the Gibbs sampling process in which these parameters appear.

The precision parameters for the Gaussian noise, i.e. $\lambda_{x_o}$ and $\lambda_{h_o}$, were initialized at $10^9$ because the data in the experiments was considered to be clean in terms
of noise. In case the noise is present in the data, lower initial values for these parameters should be chosen. Following the common practice in using the weighted Beta-Bernoulli Process, we let $\lambda_{\psi o} = 1/C$ and chose $\lambda_{\psi o}^c = \lambda_{\psi o}^t = 1$. For the Bernoulli distribution parameters $\pi_k^c, \forall c, \forall k$, we chose a non-informative initial value of 0.5. We also experimented with multiple different initial values for these parameters in the open interval $(0,1)$. The performance of our approach did not change significantly for these values. This also indicates good mixing of the underlying sampler for the Bayesian inference. We have already shown that the Gibbs sampler achieves effective mixing for the discriminative dictionaries learned under the Beta-Bernoulli Process in [7]. In our experiments, we chose the value $L/10$ for the scaling parameter of the Gaussian Process kernel $\eta_o$. This value was empirically selected and it generally resulted in good performance for all the data sets.

To choose the values of the Beta distribution parameters $a_o$ and $b_o$, we considered Lemma 8 and the discussion in Section 12.3.3 to set $a_o = b_o = \min_c |\mathcal{I}_c|$. By forcing $a_o = b_o$ we let the resulting dictionary size to be roughly close to the training data size. It was observed in [7] that such dictionary sizes generally result in the best classification performance. The value of the parameters (i.e. $\delta = \min \limits_c |\mathcal{I}_c|$) decides the rate at which the dictionary is allowed to converge to the desired size. In Fig. 12.7, we plot the dictionary size for the Salinas Image as a function of the Gibbs sampling iterations (first 100) for different values of the parameters. As can be seen, the dictionary converges slower to its final size for the larger parameter values. It was observed in our experiments that five hundred iterations of the sampler were generally enough for learning a suitable discriminative dictionary with $a_o = b_o = \min \limits_c |\mathcal{I}_c|$, using the initialization process introduced in Section 12.3.2. Therefore, we performed as many iterations of the sampler in each experiment. Our
approach required an average time of 2.1ms, 0.97ms and 5.5ms for predicting the label of a single pixel of the Indian Pines, Salinas and the Pavia University image, respectively. These timings were roughly similar for DBDL‡ due to similar dictionary sizes, whereas LC-KSVD‡ and D-KSVD‡ required around 80% of this time because of the smaller dictionaries. The timings are for Intel processor at 3.4GHz with 8GB RAM.

12.6 Conclusion

We proposed a joint representation model for learning a non-parametric discriminative Bayesian dictionary and a classifier that was suitable for hyperspectral classification. The model placed Gaussian Process priors over the dictionary atoms to incorporate the well-known relative smoothness of the natural spectra into the basis. It employed two Beta-Bernoulli processes for learning the dictionary and the classifier simultaneously, such that, the processes were coupled under the same Bernoulli distributions. This coupling benefited in classifying a test spectra when we first represented it over the dictionary and predicted its label using the resulting representation with the classifier. To exploit the contextual information in the images, we computed these representations by solving a simultaneous sparse optimization problem. We derived Gibbs sampling equations for our model and tested the approach on benchmark images for hyperspectral classification. Our experiments showed that the proposed approach consistently achieved better performance than the existing state-of-the-art discriminative dictionary learning approaches.
CHAPTER 13

Conclusion

This dissertation presented novel techniques and algorithms to address three fundamental problems in hyperspectral imaging, namely hyperspectral unmixing, super-resolution and classification. The proposed techniques exploited sparse representation framework, grounded in optimization and Bayesian theory. The proposed algorithms were systematically derived by capitalizing on the physical attributes of the hyperspectral signals and the solution objectives, which generally lead to state-of-the-art performance. Whereas the thesis contribution in the key areas of hyperspectral image analysis encourages the wide-spread use of this modality in general, few specific note-worthy research directions also originate from the findings in the dissertation. These directions are discussed first, followed by a general discussion on the future directions in hyperspectral imaging.

Novel methods for sparse unmixing of hyperspectral data were developed in the dissertation. In this regard, an important result was presented in Chapter 3 that a dictionary comprising $\ell_1$-normalized spectra always results in the representations that follow the Abundance Sum-to-one Constraint (ASC) in hyperspectral unmixing. This result can allow sparse unmixing to become more efficient by ignoring the explicit imposition of ASC in the algorithms. Computationally efficient sparse unmixing that algorithmically exploits this result is an interesting future direction to explore. The algorithms developed for this purpose can additionally take advantage of contextual information in hyperspectral images, e.g. by solving simultaneous sparse optimization problem [199], which remained unexplored in this thesis.

For the problem of sparse unmixing, it was generally observed that supervised unmixing methods performed well in synthetic settings. However, their performance was compromised in real settings where it was hard to guarantee that the a priori known dictionary contained all the endmembers present in the image. Critically speaking, ensuring the presence of the relevant endmembers in an a priori known dictionary is already solving half of the endmember identification problem. Therefore, in the area of sparse unmixing, there is a strong need of methods that can identify useful dictionaries for the supervised sparse unmixing techniques. In comparison to the popularity of sparse unmixing approaches, the existing literature seriously lacks in this direction of research.

For the problem of hyperspectral super-resolution, an evaluation protocol was followed in the experiments of Chapter 6, 7 and 8 that synthetically generated the low resolution hyperspectral image by down sampling the ground truth. Although
this protocol was found useful in quantitative assessment of the results, it did not strictly represent practical conditions. In practice, the images being fused may suffer from alignment mismatches and other physical constraints. These factors could not be catered for in the experimental results because no existing hyperspectral image database provides the high resolution ground truth along real hyperspectral images. Keeping in view the recent interest of researchers in the image fusion based hyperspectral super-resolution, this thesis identifies a strong need of benchmark database(s) for the hyperspectral super-resolution problem. Such a database should provide a set of hyperspectral and RGB images of a scene acquired under practical conditions along, either a high resolution hyperspectral image of the same scene, or spectral response of the scene at multiple locations as a gold standard.

In Chapter 7, a Bayesian sparse coding strategy was proposed. Similarly, spatial and spectral kernels were introduced in a Bayesian sparse representation model and its inference scheme was proposed in Chapter 8. These methods were used for hyperspectral super-resolution in this thesis, nevertheless they are generic methods that allow the use of prior domain knowledge in a principled manner due to their Bayesian nature. Considering the recent success of the optimization based sparse representation framework in numerous computer vision tasks, exploitation of appropriate prior knowledge for these tasks using the proposed Bayesian methods is another interesting research direction originating from this dissertation.

Chapter 9 of the thesis established the importance of sparsity in collaborative representation based classification. Moreover, it showed that dense representations can be combined with the sparse ones for improved classification. Finding the optimal method for combining the representations for this purpose is an interesting area that can yield interesting results in the future. Similarly, Chapters 10-12 use Beta Process [157] for discriminative Bayesian dictionaries. Although the developed approaches achieve good performance, theoretically, they are not designed for infinite partitions of the measurable space, which is often expected from the non-parametric Bayesian methods. It will be an interesting theoretical research to extend these approaches with the help of Hierarchical Beta Process [193] for potentially infinite amount of training data and infinite dictionaries.

Overall, the thesis has developed a wide variety of methods for the fundamental problems in hyperspectral imaging. These methods ranged from simple linear approaches to more complex non-parametric Bayesian methods. In the authors experience, simpler approaches that rely on linear modeling (e.g. in unmixing and super-resolution) generally provide a good baseline performance and they are also
computationally efficient. Nevertheless, there are two major limitations of such approaches for hyperspectral image analysis. Firstly, they heavily rely on the chosen parameter values. Normally, the parameter values can be optimized using cross-validation. However, it is generally the case in hyperspectral imaging that the available training data is very limited, if available at all. This causes problems in finding optimal values of parameters using the cross-validation approach. Secondly, in many hyperspectral imaging problems some important features of the spectral signals of interest are known a prior. The known features can be exploited for improved performance by incorporating prior knowledge in the models and techniques. Generally, it is hard to incorporate such prior knowledge in simpler models and techniques in a principled manner. More complex models easily overcome these problems. For instance, the Bayesian models developed in the second and the third part of the thesis are highly insensitive to the chosen parameter values (as they adjust these values according to the data themselves). They also incorporate spatial and spectral smoothness of the signals in the models in a principled manner, thereby achieving better performance. However, it should be noted that the improved performance comes at an extra computational cost of Bayesian inference. Therefore, the ideal scenarios of using more complex techniques are where the processing of the data can be done off-line.

Hyperspectral imaging is becoming increasingly popular in computer vision applications. Whereas most of the existing hyperspectral cameras used in the mainstream computer vision research sample the electromagnetic spectrum in the visible range in 30 - 33 channels, these channels are expected to increase rapidly and the sensing is expected to expand beyond the visible spectrum range. This can be anticipated because the existing hyperspectral instruments installed on many remote sensing platforms are already sampling the electromagnetic spectrum beyond the visible range at hundreds of wavelength intervals. We can find definitive evidence in the recent literature that hyperspectral imaging improves performance of many fundamental computer vision tasks, as compare to RGB imaging. With the advancement of hardware technology, we can expect an exponential increase in the use of hyperspectral imaging in the main stream computer vision tasks. Therefore, the generic techniques to solve the fundamental problems of hyperspectral imaging developed in the thesis are anticipated to produce a high impact in the future computer vision research.
Here we show that the results from theorem (3.1) in Section 3.2.3 are in line with Elad’s results in [31]. We follow the line of reasoning of [31] and arrive at the condition of equivalency of these results.

**Proof:** Consider the model

\[ D\alpha = y, \ s.t. \ \alpha > 0. \]  \hspace{1cm} (A.1)

Since all the entries in \( D \) (i.e. reflectances) are non-negative, the row span of \( D \) intersects the positive orthant. Mathematically,

\[ \exists \ h \ s.t. \ h^T D = w^T > 0. \]  \hspace{1cm} (A.2)

Let \( W = \text{diag}(w) \) (strictly positive definite), then (A.1) can be re-written as:

\[ DW^{-1}W\alpha = y, \ s.t. \ \alpha > 0. \]  \hspace{1cm} (A.3)

Or

\[ \hat{D}z = y, \ z > 0. \]  \hspace{1cm} (A.4)

where \( \hat{D} = DW^{-1} \) and \( z = W\alpha \). Left multiplying Equation (A.1) by \( h^T \) on both sides gives \( c = h^T y \) on the right hand side. Since \( h^T D = w^T \), we have \( w^T \alpha = c \).

That means:

\[ w^T \alpha = 1^T W\alpha = 1^T z = c \]  \hspace{1cm} (A.5)

where \( 1 \in \mathbb{R}^m \) is a column vector of 1s. Equation (A.5) suggests that the \( l_1 \)-norm of \( z \in \{ \hat{D}z = y, \ z > 0 \} \) is the constant \( c \).

In the above equations we can always choose \( h = 1 \), which makes \( \hat{D} \) nothing but \( D \) with its columns normalized in \( l_1 \)-norm and makes \( c = ||y||_1 \). Thus, if the columns of \( D \) already have unit \( l_1 \)-norm then \( W \) is an identity matrix and \( z = \alpha \). Therefore, \( ||z||_1 = ||\alpha||_1 = ||y||_1 \). This, in turn, implies automatic imposition of the generalized ASC, where \( ||y||_1 \) becomes the pixel-dependent scale factor. Note that, this result coincides with Corollary 2a in Section (3.2.3) and we arrive here after assuming the normalized version of \( D \).

It is easy to see that there can be many potential \( h \) vectors that satisfy the condition in (A.2). Which implies the existence of many potential matrices for \( W \), and many potential values of \( c \). Therefore, \( c \) becomes a constant only when a particular \( h \) is operated on \( D \) to get \( w \). In [31], Elad used \( h = 1 \) to convert the systems of equations in (A.1) to (A.4) and claimed \( ||z||_1 \) to be constant. No claim for \( ||\alpha||_1 \) to be constant was made in [31].
APPENDIX B

B.1 Derivation of the Gibbs Sampling Equations

B.1.1 The Model

Let \( y_i \in \mathbb{R}^L, \Phi \in \mathbb{R}^{L \times |K|}, \beta_i \in \mathbb{R}^{|K|} \) and \( \epsilon_i \in \mathbb{R}^L \). Then \( \forall i \in \{1, ..., mn\} \) and \( \forall k \in \mathcal{K} = \{1, ..., K\} \):

\[
\begin{align*}
    y_i & = \Phi \beta_i + \epsilon_i \\
    \beta_i & = z_i \odot s_i \\
    \varphi_k & \sim \mathcal{N}(\varphi_k | \mu_k, \Lambda_k^{-1}) \\
    z_{ik} & \sim \text{Bern}(z_{ik} | \pi_k) \\
    \pi_k & \sim \text{Beta}(\pi_k | a_o/K, b_o/(K - 1)/K) \\
    s_{ik} & \sim \mathcal{N}(s_{ik} | \mu_s, \lambda_s^{-1}) \\
    \epsilon_i & \sim \mathcal{N}(\epsilon_i | 0, \Lambda \epsilon_o^{-1}).
\end{align*}
\]

For computational purposes, let \( \mu_s = 0, \mu_k = 0, \Lambda_k = \lambda_k I_L \) and \( \Lambda \epsilon_o = \lambda_o I_L \), where \( I_L \in \mathbb{R}^{L \times L} \) is an identity matrix. We further place the following priors on the precision parameters of the normal distributions:

\[
\lambda_s \sim \Gamma(\lambda_s | c_o, d_o), \lambda \sim \Gamma(\lambda | e_o, f_o).
\]

We make use of the following theorem [26] in our proves:

**Theorem B.1** [26]: When prior distribution over \( x_1 \) is given by \( p(x_1) = \mathcal{N}(x_1 | \mu_o, \Lambda_o^{-1}) \) and the likelihood function is defined as \( p(x_2 | x_1) = \mathcal{N}(x_2 | Ax_1 + b, L^{-1}) \), the posterior distribution over \( x_1 \) can be written as \( p(x_1 | x_2) = \mathcal{N}(x_1 | \mu, \Lambda^{-1}) \), where

\[
\begin{align*}
    \Lambda & = \Lambda_o + A^T L A \\
    \mu & = \Lambda^{-1}(A^T L (y - b) + \Lambda_o \mu_o).
\end{align*}
\]

B.1.2 Gibbs Sampling Equations

**Sample \( \varphi_k \):** From our model, we can write the posterior distribution over the dictionary atom \( p(\varphi_k | -) \) as:

\[
p(\varphi_k | -) \propto \prod_{i=1}^{mn} \mathcal{N}(y_i | \Phi(z_i \odot s_i), \lambda_{k_o}^{-1} I_L) \mathcal{N}(\varphi_k | 0, \lambda_{k_o}^{-1} I_L).
\]

In order to write the mean parameter of the likelihood function in terms of \( \varphi_k \), we can write:

\[
y_{i \varphi_k} = y_i - \Phi(z_i \odot s_i) + \varphi_k(z_{ik} \odot s_{ik}).
\]
where \( y_{i\varphi_k} \) represents the contribution of the dictionary atom \( \varphi_k \) to the signal \( y_i \). This gives us the following form of the posterior distribution over \( \varphi_k \):

\[
p(\varphi_k|\cdot) \propto \prod_{i=1}^{mn} N(y_{i\varphi_k}|\varphi_k (z_{ik} s_{ik}), \lambda_{\epsilon o}^{-1} I_L) N(\varphi_k|0, \lambda_{\epsilon o}^{-1} I_L).
\]

Using the results of Theorem B.1, the posterior distribution over the dictionary atoms is given by:

\[
p(\varphi_k|\cdot) = N(\varphi_k|\mu_k, \lambda_k^{-1} I_L), \tag{B.1}
\]

where

\[
\lambda_k = \lambda_{\epsilon o} + \lambda_{\epsilon o} \sum_{i=1}^{mn} (z_{ik} s_{ik})^2, \quad \mu_k = \lambda_{\epsilon o}^{-1} \lambda_{\epsilon o} \sum_{i=1}^{mn} (z_{ik} s_{ik}) y_{i\varphi_k}.
\]

Note that, we arrive at the above results by putting \( A = \sum_{i=1}^{mn} (z_{ik} s_{ik}) \) and \( b = 0 \) in the results of Theorem B.1. The results get further simplified because of the standard multivariate Gaussian distributions with isotropic covariance/precision matrices.

**Sample \( z_{ik} \):** Once \( \varphi_k \) has been sampled, we must sample \( z_{ik} \) (and \( s_{ik} \)) based on the updated atom. Again, using the contribution of the \( k^{th} \) dictionary atom only, the posterior over \( z_{ik} \) can be written as:

\[
p(z_{ik}|\cdot) \propto N(y_{i\varphi_k}|\varphi_k (z_{ik} s_{ik}), \lambda_{\epsilon o}^{-1} I_L) Bern(z_{ik}|\pi_{\epsilon o}).
\]

Thus,

\[
p(z_{ik} = 1|\cdot) \propto \pi_{\epsilon o} \exp \left( - \frac{(y_{i\varphi_k} - \varphi_k s_{ik})^T \lambda_{\epsilon o} I_L (y_{i\varphi_k} - \varphi_k s_{ik})}{2} \right)
\]

\[
= \pi_{\epsilon o} \exp \left( - \frac{\lambda_{\epsilon o}}{2} (y_{i\varphi_k} y_{i\varphi_k}^T \varphi_k s_{ik}^2 + \varphi_k^T \varphi_k s_{ik}^2) \right)
\]

\[
p_1 = \pi_{\epsilon o} \exp \left( - \frac{\lambda_{\epsilon o}}{2} y_{i\varphi_k} y_{i\varphi_k}^T \right) \exp \left( - \frac{\lambda_{\epsilon o}}{2} \varphi_k s_{ik}^2 \right)
\]

Similarly,

\[
p(z_{ik} = 0|\cdot) \propto (1 - \pi_{\epsilon o}) \exp \left( - \frac{y_{i\varphi_k}^T \lambda_{\epsilon o} I_L y_{i\varphi_k}}{2} \right)
\]

\[
= (1 - \pi_{\epsilon o}) \exp \left( - \frac{\lambda_{\epsilon o}}{2} y_{i\varphi_k}^T y_{i\varphi_k} \right)
\]

\[
p_0 = \exp \left( - \frac{\lambda_{\epsilon o}}{2} y_{i\varphi_k} y_{i\varphi_k}^T \right) - \pi_{\epsilon o} \exp \left( - \frac{\lambda_{\epsilon o}}{2} y_{i\varphi_k} y_{i\varphi_k} \right)
\]

Thus, \( z_{ik} \) can be sampled from the following normalized Bernoulli distribution.

\[
z_{ik} \sim Bern \left( \frac{p_1}{p_1 + p_0} \right).
\]
Further simplification leads to the following formulation:

\[ z_{ik} \sim \text{Bern}\left( \frac{\pi_k \xi}{1 - \pi_k + \xi \pi_k} \right). \]  \hspace{1cm} (B.2)

where, \( \xi = \exp\left( -\frac{\lambda_s}{2} (\varphi_k^T \varphi_k s_{ik}^2 - 2s_{ik} y_{i_{k}}^T \varphi_k) \right). \)

**Sample \( s_{ik} \):** With the same reasoning as above, we can write the following about the posterior distribution over \( s_{ik} \):

\[ p(s_{ik}|-) \propto \mathcal{N}(y_{i_{k}}|\varphi_k(z_{ik},s_{ik}), \lambda_s^{-1} I_L)N(s_{ik}|0, \lambda_s^{-1}). \]

Using Theorem B.1, \( s_{ik} \) can be sampled from the following distribution:

\[ p(s_{ik}|-) = \mathcal{N}(s_{ik}|\mu_s, \lambda_s^{-1}), \]  \hspace{1cm} (B.3)

where,

\[ \lambda_s = \lambda_s + (\varphi_k z_{ik})^T \lambda_s I_L (\varphi_k z_{ik}) \]
\[ = \lambda_s + \lambda_s z_{ik}^2 \varphi_k^T \varphi_k. \]
\[ \mu_s = \lambda_s^{-1} (\varphi_k z_{ik})^T \lambda_s I_L y_{i_{k}} \]
\[ = \lambda_s^{-1} \lambda_s z_{ik} \varphi_k^T y_{i_{k}}. \]

**Sample \( \pi_k \):** Using the model, the posterior distribution over \( \pi_k \) can be written as:

\[ p(\pi_k|-) \propto \prod_{i=1}^{mn} \text{Bern}(z_{ik}|\pi_k) \text{Beta}(\pi_k|a_o/K, b_o(K-1)/K) \]
\[ = \pi_k^{\sum_{i=1}^{mn} z_{ik}} (1 - \pi_k)^{mn - \sum_{i=1}^{mn} z_{ik}} \times \frac{\pi_k^{-1}}{\lambda_s} (1 - \pi_k)^{-1} \]
\[ = \pi_k^{a_o + \sum_{i=1}^{mn} z_{ik}^{-1}} \frac{b_o(K-1)}{K} + mn - \sum_{i=1}^{mn} z_{ik}^{-1} \]
\[ = \text{Beta}(\frac{a_o}{K} + \sum_{i=1}^{mn} z_{ik}, \frac{b_o(K-1)}{K} + mn - \sum_{i=1}^{mn} z_{ik}). \]

Thus,

\[ \pi_k \sim \text{Beta}\left(\frac{a_o}{K} + \sum_{i=1}^{mn} z_{ik}, \frac{b_o(K-1)}{K} + mn - \sum_{i=1}^{mn} z_{ik}\right). \]  \hspace{1cm} (B.4)

**Sample \( \lambda_s \):** In the model, we have assumed that \( \mu_s = 0 \) and \( \lambda_{sk} = \lambda_s, \forall k \in K \).

This simplification allows us to write the likelihood function for \( \lambda_s \) in terms of
standard multivariate Gaussian with isotropic covariance matrix $$\lambda_{s_{o}}^{-1}I_{K}$$. Thus, we can write the posterior over $$\lambda_{s}$$ as following:

$$p(\lambda_{s} | -) \propto \prod_{i=1}^{mn} \mathcal{N}(s_{i} | 0, \lambda_{s_{o}}^{-1}I_{K}) \Gamma(\lambda_{s_{o}} | c_{o}, d_{o})$$

where $$\Gamma(.)$$ is the gamma function and $$|.|$$ denotes the determinant of the matrix. Neglecting the constants and making use of the property that $$|\lambda I_{K}| = \lambda^{K}$$, we arrive at the following:

$$p(\lambda_{s} | -) \propto \lambda_{s_{o}}^{\frac{Kmn}{2}} \exp \left( - \frac{\lambda_{s_{o}}}{2} \sum_{i=1}^{mn} s_{i}^{T}s_{i} \right) \lambda_{s_{o}}^{\frac{c_{o} - 1}{2}} \exp(-d_{o} \lambda_{s_{o}})$$

Therefore,

$$\lambda_{s} \sim \Gamma(\frac{Kmn}{2} + c_{o}, \frac{1}{2} \sum_{i=1}^{mn} \|s_{i}\|^{2} + d_{o}). \quad (B.5)$$

**Sample $$\lambda_{c}$$**: Based on our model, we can write the posterior over $$\lambda_{c}$$ as:

$$p(\lambda_{c} | -) \propto \prod_{i=1}^{mn} \mathcal{N}(y_{i} | \Phi(z_{i} \odot s_{i}), \lambda_{c_{o}}^{-1}I_{L}) \Gamma(\lambda_{c_{o}} | e_{o}, f_{o}).$$

Similar to $$\lambda_{s}$$ we can arrive at the following for sampling $$\lambda_{c}$$:

$$\lambda_{c} \sim \Gamma(\frac{Lmn}{2} + e_{o}, \frac{1}{2} \sum_{i=1}^{mn} \|y_{i} - \Phi(z_{i} \odot s_{i})\|^{2} + f_{o}). \quad (B.6)$$

### B.2 MSE Analysis for Lemma 1

We can write $$MSE$$ as below:

$$MSE = \mathbb{E} \left[ \|\tilde{\alpha} - \alpha\|^{2} \right]. \quad (B.7)$$

In our settings, we can re-write the above definition as:

$$MSE = \sum_{z \in \mathcal{Z}} P(z) \mathbb{E} \left[ \|\tilde{\alpha} - \alpha\|^{2} | z \right]. \quad (B.8)$$
Below, we further analyze the expression for \( MSE \), starting from expanding the second term in the above equation.

\[
\mathbb{E}[||\hat{\alpha} - \alpha||_2^2 | z] = \mathbb{E}[||\hat{\alpha}||_2^2 - 2\hat{\alpha}^T\alpha + ||\alpha||_2^2 | z] 
= ||\hat{\alpha}||_2^2 - 2\hat{\alpha}^T\mathbb{E}[\alpha|z] + \mathbb{E}[||\alpha||_2^2 | z].
\]

(B.9)

(B.10)

We can further write \( \mathbb{E}[||\alpha||_2^2 | z] \) as:

\[
\mathbb{E}[||\alpha||_2^2 | z] = \mathbb{E}[||\mathbb{E}[\alpha|z] + \alpha - \mathbb{E}[\alpha|z]||_2^2 | z]
= \mathbb{E}[||\mathbb{E}[\alpha|z]||_2^2 + 2\mathbb{E}[\alpha|z](\alpha - \mathbb{E}[\alpha|z]) + ||\alpha - \mathbb{E}[\alpha|z]||_2^2 | z]
= \mathbb{E}[||\mathbb{E}[\alpha|z]||_2^2 + 2\mathbb{E}[\alpha|z] - 2||\mathbb{E}[\alpha|z]||_2^2 + ||\alpha - \mathbb{E}[\alpha|z]||_2^2 | z]
= ||\mathbb{E}[\alpha|z]||_2^2 + 2||\mathbb{E}[\alpha|z]||_2^2 - 2||\mathbb{E}[\alpha|z]||_2^2 + \mathbb{E}[||\alpha - \mathbb{E}[\alpha|z]||_2^2 | z]
= ||\mathbb{E}[\alpha|z]||_2^2 + \mathbb{E}[||\alpha - \mathbb{E}[\alpha|z]||_2^2 | z].
\]

Denoting the second term in the above equation as \( V_z \) (i.e. conditional variance) and putting the values in (B.10) gives:

\[
\mathbb{E}[||\hat{\alpha} - \alpha||_2^2 | z] = ||\hat{\alpha}||_2^2 - 2\hat{\alpha}^T\mathbb{E}[\alpha|z] + ||\mathbb{E}[\alpha|z]||_2^2 + V_z
= ||\hat{\alpha} - \mathbb{E}[\alpha|z]||_2^2 + V_z.
\]

(B.11)

(B.12)

Putting the values back in (B.8).

\[
MSE = \sum_{z \in Z} P(z)||\hat{\alpha} - \mathbb{E}[\alpha|z]||_2^2 + \sum_{z \in Z} P(z)V_z
= \mathbb{E}[||\hat{\alpha} - \mathbb{E}[\alpha|z]||_2^2] + \mathbb{E}[V_z].
\]

(B.13)

(B.14)

To find the minimizer, we derivate \( MSE \) w.r.t \( \hat{\alpha} \) and equate it to 0.

\[
0 = \mathbb{E}\left[ \frac{\partial}{\partial \hat{\alpha}} ||\hat{\alpha} - \mathbb{E}[\alpha|z]||_2^2 \right] + \mathbb{E}\left[ \frac{\partial}{\partial \hat{\alpha}} V_z \right]
= \mathbb{E}\left[ \left( \frac{\hat{\alpha} - \mathbb{E}[\alpha|z]}{\sqrt{||\hat{\alpha} - \mathbb{E}[\alpha|z]||_2^2}} \right) \cdot \frac{\mathbb{E}[\alpha|z] - \mathbb{E}[\alpha|z]}{||\hat{\alpha} - \mathbb{E}[\alpha|z]||_2^2} \right]
= \hat{\alpha} - \mathbb{E}[\mu_z(\alpha)].
\]

(B.15)

(B.16)

(B.17)

where, we have denoted \( \mathbb{E}[\alpha|z] \) as \( \mu_z(\alpha) \). Clearly from (B.17) \( \hat{\alpha}_{opt} = \mathbb{E}[\mu_z(\alpha)] \).
C.1 Further Ground-based Image Results

Harvard Database [40] Images: Table C.1 summarizes the results on the Harvard database [40]. We report the results on the images with stationary scenes only, to avoid any reconstruction bias. For comparison, results of the existing state-of-the-art approach BSR [5] are also provided. Spectral reconstructions of four representative images are also provided below.

Table C.1: Results on Harvard database [40]: The results are in the range of 8 bit images. Blue cells indicate that spectral reconstructions are also shown below.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RMSE</td>
<td>Sam</td>
<td></td>
<td></td>
<td>RMSE</td>
<td>Sam</td>
</tr>
<tr>
<td>1</td>
<td>Img h0</td>
<td>2.2</td>
<td>2.9</td>
<td>12</td>
<td>Img d3</td>
<td>1.4</td>
<td>3.2</td>
</tr>
<tr>
<td>2</td>
<td>Img h1</td>
<td>0.6</td>
<td>0.8</td>
<td>13</td>
<td>Img d2</td>
<td>0.8</td>
<td>3.7</td>
</tr>
<tr>
<td>3</td>
<td>Img h2</td>
<td>0.7</td>
<td>0.5</td>
<td>14</td>
<td>Img c5</td>
<td>0.9</td>
<td>2.0</td>
</tr>
<tr>
<td>4</td>
<td>Img h3</td>
<td>0.8</td>
<td>2.4</td>
<td>15</td>
<td>Img c2</td>
<td>2.4</td>
<td>2.6</td>
</tr>
<tr>
<td>5</td>
<td>Img f5</td>
<td>0.7</td>
<td>1.1</td>
<td>16</td>
<td>Img b9</td>
<td>1.2</td>
<td>0.9</td>
</tr>
<tr>
<td>6</td>
<td>Img f4</td>
<td>0.7</td>
<td>0.7</td>
<td>17</td>
<td>Img b5</td>
<td>0.8</td>
<td>2.1</td>
</tr>
<tr>
<td>7</td>
<td>Img e6</td>
<td>1.3</td>
<td>1.3</td>
<td>18</td>
<td>Img b2</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>8</td>
<td>Img e4</td>
<td>0.6</td>
<td>0.8</td>
<td>19</td>
<td>Img b1</td>
<td>0.8</td>
<td>1.5</td>
</tr>
<tr>
<td>9</td>
<td>Img d9</td>
<td>1.3</td>
<td>2.7</td>
<td>20</td>
<td>Img a7</td>
<td>0.8</td>
<td>1.7</td>
</tr>
<tr>
<td>10</td>
<td>Img d7</td>
<td>0.9</td>
<td>3.0</td>
<td>21</td>
<td>Img a5</td>
<td>0.3</td>
<td>1.9</td>
</tr>
<tr>
<td>11</td>
<td>Img d4</td>
<td>0.7</td>
<td>3.6</td>
<td>22</td>
<td>Img 1</td>
<td>0.8</td>
<td>2.0</td>
</tr>
</tbody>
</table>
### C.1. Further Ground-based Image Results

<table>
<thead>
<tr>
<th>Harvard Img d3</th>
<th>Harvard Img b5</th>
<th>Harvard Img c2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low resolution spectral image</td>
<td>Ground truth</td>
<td>Super resolution spectral image</td>
</tr>
<tr>
<td>460 nm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>540 nm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>630 nm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Absolute difference (Proposed)</td>
<td>Absolute difference (BSR)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Colors represent the magnitude of the absolute difference, with blue indicating a larger difference.
CAVE Database [226] Images: Quantitative results on CAVE database [226] are given in Table C.2. For qualitative analysis, illustrations of spectral reconstructions of four representative images are also shown below.

Table C.2: Results on CAVE database [226]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Balloons</td>
<td>1.9</td>
<td>7.6</td>
<td>16</td>
<td>Faces</td>
<td>1.9</td>
<td>15.3</td>
</tr>
<tr>
<td>2</td>
<td>Beads</td>
<td>5.8</td>
<td>13.7</td>
<td>17</td>
<td>Photo and Face</td>
<td>1.7</td>
<td>10.5</td>
</tr>
<tr>
<td>3</td>
<td>CD</td>
<td>5.3</td>
<td>10.6</td>
<td>18</td>
<td>Hairs</td>
<td>1.5</td>
<td>10.0</td>
</tr>
<tr>
<td>4</td>
<td>Cloth</td>
<td>3.7</td>
<td>5.0</td>
<td>19</td>
<td>Oil Painting</td>
<td>2.7</td>
<td>8.0</td>
</tr>
<tr>
<td>5</td>
<td>Clay</td>
<td>3.0</td>
<td>10.1</td>
<td>20</td>
<td>Paints</td>
<td>3.1</td>
<td>10.3</td>
</tr>
<tr>
<td>6</td>
<td>Statue</td>
<td>1.2</td>
<td>18.3</td>
<td>21</td>
<td>Water color</td>
<td>1.9</td>
<td>4.7</td>
</tr>
<tr>
<td>7</td>
<td>Feathers</td>
<td>5.0</td>
<td>15.7</td>
<td>22</td>
<td>Beers</td>
<td>2.0</td>
<td>2.6</td>
</tr>
<tr>
<td>8</td>
<td>Flowers</td>
<td>4.9</td>
<td>13.2</td>
<td>23</td>
<td>Jelly Beans</td>
<td>4.6</td>
<td>11.8</td>
</tr>
<tr>
<td>9</td>
<td>Glass Tiles</td>
<td>2.8</td>
<td>8.3</td>
<td>24</td>
<td>Lemon Slices</td>
<td>2.0</td>
<td>12.3</td>
</tr>
<tr>
<td>10</td>
<td>Chart &amp; Toys</td>
<td>3.3</td>
<td>11.1</td>
<td>25</td>
<td>Lemons</td>
<td>1.7</td>
<td>8.8</td>
</tr>
<tr>
<td>11</td>
<td>Pompons</td>
<td>3.9</td>
<td>10.1</td>
<td>26</td>
<td>Peppers</td>
<td>2.2</td>
<td>10.1</td>
</tr>
<tr>
<td>12</td>
<td>Sponges</td>
<td>5.1</td>
<td>8.1</td>
<td>27</td>
<td>Strawberries</td>
<td>2.1</td>
<td>7.6</td>
</tr>
<tr>
<td>13</td>
<td>Spools</td>
<td>4.7</td>
<td>14.9</td>
<td>28</td>
<td>Sushi</td>
<td>2.9</td>
<td>12.7</td>
</tr>
<tr>
<td>14</td>
<td>Stuffed Toys</td>
<td>4.5</td>
<td>13.2</td>
<td>29</td>
<td>Tomatoes</td>
<td>2.2</td>
<td>8.0</td>
</tr>
<tr>
<td>15</td>
<td>Super Balls</td>
<td>2.4</td>
<td>15.8</td>
<td>30</td>
<td>Yellow Peppers</td>
<td>2.1</td>
<td>10.8</td>
</tr>
</tbody>
</table>

![Image of spectral reconstructions](image_url)
C.1. Further Ground-based Image Results
C.2 Further AVIRIS Image [80] Results

The images are provided below. For comparison, figures also include the absolute difference between the ground truths and the reconstructions for the second best results. For SC02, these results were obtained by GSOMP [8], whereas BSR [5] achieved these results on the remaining images.
### C.2. Further AVIRIS Image Results

<table>
<thead>
<tr>
<th>AVIRIS Image SC03</th>
<th>Low resolution spectral images</th>
<th>Ground truth</th>
<th>Super resolution spectral images (Proposed)</th>
<th>Absolute difference (BSR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>460 nm</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>540 nm</td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
<tr>
<td>620 nm</td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
<tr>
<td>1500 nm</td>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
<td><img src="image15.png" alt="Image" /></td>
<td><img src="image16.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AVIRIS Image SC04</th>
<th>Low resolution spectral images</th>
<th>Ground truth</th>
<th>Super resolution spectral images (Proposed)</th>
<th>Absolute difference (BSR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>460 nm</td>
<td><img src="image17.png" alt="Image" /></td>
<td><img src="image18.png" alt="Image" /></td>
<td><img src="image19.png" alt="Image" /></td>
<td><img src="image20.png" alt="Image" /></td>
</tr>
<tr>
<td>540 nm</td>
<td><img src="image21.png" alt="Image" /></td>
<td><img src="image22.png" alt="Image" /></td>
<td><img src="image23.png" alt="Image" /></td>
<td><img src="image24.png" alt="Image" /></td>
</tr>
<tr>
<td>620 nm</td>
<td><img src="image25.png" alt="Image" /></td>
<td><img src="image26.png" alt="Image" /></td>
<td><img src="image27.png" alt="Image" /></td>
<td><img src="image28.png" alt="Image" /></td>
</tr>
<tr>
<td>1500 nm</td>
<td><img src="image29.png" alt="Image" /></td>
<td><img src="image30.png" alt="Image" /></td>
<td><img src="image31.png" alt="Image" /></td>
<td><img src="image32.png" alt="Image" /></td>
</tr>
</tbody>
</table>
C.3 Robustness to Misalignment

Our approach assumes aligned hyperspectral and multi-spectral images following the standard practice. However, it extracts a scene’s material spectra using the hyperspectral image $X^h$ and imposes them over the scene’s spatial patterns, captured by the multi-spectral image $X$. Hence, unless the misalignment between the images is so severe that $X^h$ does not image the same materials as imaged by $X$, the performance of our approach does not degrade significantly. To exemplify, we show the RMSE for 3 random CAVE images [226] against different offsets ($\delta$). In Fig. C.1, $\delta = n$ means simultaneous misalignment in both spatial dimensions by $n$-pixels. We used 256 × 256 central image patches here. A graceful degradation of the performance is evident from the figure.

C.4 Illustration of Blur in CAVE Image [226]

(Left) Ground truth spectral image for ‘Oil Painting’. (Right) Reconstructed spectral image by the proposed approach. To avoid any bias in the results introduced by the blur in the ground truth, we do not use the first two channels (at 400 and 410nm) of the images in our experiments.
D.1 Dictionary Visualization

In chapter 10, the training samples for our experiments consisted of random face features [213] for face recognition, spatial pyramid features [121] for object and scene classification; and action bank features [179] for action recognition. Since the atoms of a learned dictionary generally resemble the training samples, appearance of the dictionary atoms did not provide useful visual information for those experiments. Nevertheless, for the sake of illustration, we show some popular dictionary atoms in the first and the third row of Fig. D.1 for the Extended YaleB dataset. Each sub-figure displays the twelve most popular dictionary atoms for a plot shown in Fig. 4 of the paper. For a more informative visualization of the popular atoms, we also conducted a parallel experiment in which we replaced the training data by the corresponding cropped images of the Extended YaleB database. Figure D.1 also includes the visualizations of the learned popular dictionary atoms for that experiment. Observing these atoms, we can conjecture that the popular dictionary atoms tend to preserve the dominant features of their relevant class data. Although, in each sub-figure, we have arranged the atoms from top-left to bottom-right with the decreasing values of $\pi_k$, the relative difference in the values of $\pi_k$ was observed to be very small for most of the displayed atoms of the same class.

D.2 Gibbs Sampling Analysis

We assessed the inference accuracy of our Gibbs sampler by analyzing the mixing and the potential chain reduction factors [75] for the key parameters of our model, i.e. $\pi^{c \in \{1, \ldots, C\} \in \mathbb{R}^{|K|}$. The most important characteristics of our model, i.e. forcing dictionary to be discriminative and inferring the desired dictionary size, are directly related to the $C \times |K|$ Bernoulli distribution parameters stored in the set $\pi^{c \in \{1, \ldots, C\}}$. In Fig. D.2, we show the trace plots, histograms of traces and the running means of two representative parameters from the set $\pi^{c \in \{1, \ldots, C\}}$ for the AR database [140]. In the figure, the values of $c$ and $k$ are chosen at random for each $\pi_k$. The behavior of the plots for the other values of $c$ and $k$ was also observed to be qualitatively similar. Therefore, we only display the representative examples. The behavior of the plots clearly indicate good mixing of the sampler. In Figure D.3, we show the representative examples for the Fifteen Scene Category [121] databases. These examples also represent the typical behavior of the plots for the data set. Again, good mixing is evident from the plots. Qualitatively similar behavior was observed for the remaining databases as well.
Figure D.1: Visualization of the most active dictionary atoms for six classes of Extended YaleB: Random face features were used as the training data for the dictionary atoms shown in the first and the third row. These atoms correspond to the largest values of $\pi_k^c$ in Fig. 4 of the paper. The second and the fourth row used cropped face images as the training data. The arrangement for the most active to the least active atom is from top-left to bottom-right in each subfigure. The class label is denoted as ‘c’.
D.2. Gibbs Sampling Analysis

To quantitatively analyze the convergence of the sampler, we followed Gelman and Rubin [75] and monitored the Potential Scale Reduction Factors (PSRFs) for the parameters $\pi_c \in \{1, \ldots, C\}$. For computing the factors, we ran 10 sampling processes for each database. For initializing $\pi_c^k$, $\forall k$, $\forall c$ in each process, we randomly drew samples from the standard uniform distribution on the open interval (0, 1). In each experiment, the processes were run for $2n$ iterations and the last $n$ iterations were used to compute the reduction factors. For the procedural details of computing PSRFs, we refer to [75]. In Fig. D.4, we plot the worst-case reduction factors for four data sets against increasing number of iterations of the employed Gibbs sampler. Here, the worst-case PSFRs are the maximum values among the $C \times |K|$ PSRFs.
Figure D.4: Worst-case Potential Scale Reduction Factor (PSRF) [75] as a function of Gibbs sampling iterations: For each dataset, the plotted $\pi^c_k$ values are the maximum among the $C \times |K|$ values after the reported number of iterations.

Table D.1: Potential Scale Reduction Factors [75] for the parameters $\pi^c \in \{1,\ldots,C\}$ for each database, after five hundred Gibbs sampling iterations. The values are computed by running 10 chains and initializing the parameters by random draws from a uniform distribution over the open interval (0,1).

<table>
<thead>
<tr>
<th>Data set</th>
<th>Maximum</th>
<th>Mean</th>
<th>Std. dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR Database</td>
<td>1.058</td>
<td>1.001</td>
<td>9.9 \times 10^{-4}</td>
</tr>
<tr>
<td>Extended YaleB</td>
<td>1.021</td>
<td>1.000</td>
<td>9.3 \times 10^{-4}</td>
</tr>
<tr>
<td>Caltech-101 (30)</td>
<td>1.023</td>
<td>1.000</td>
<td>5.1 \times 10^{-4}</td>
</tr>
<tr>
<td>Fifteen Scene Category</td>
<td>1.012</td>
<td>1.000</td>
<td>7.6 \times 10^{-4}</td>
</tr>
</tbody>
</table>

values for a set $\pi^c \in \{1,\ldots,C\}$. In the figure, these values become very close to 1 after five hundred iterations of the sampler for each dataset. Since the shown values are for the worst cases, we can conjecture that the performed Gibbs sampling infers the posterior distributions fairly accurately after five hundred iterations. In Table D.1, we also report the mean and the standard deviations of the $C \times |K|$ PSRF values for each dataset, achieved after five hundred iterations of the sampler. These values suggest consistent convergence of PSRFs to 1 for almost all the parameters.

D.3 Performance Dependence on Training Data Size

A larger amount of training data generally results in better classification performance of the proposed approach. Figure D.5 plots classification accuracy of our
D.3. Performance Dependence on Training Data Size

Figure D.5: Classification accuracy of the proposed approach as a function of training data size. AR database [140] and Extended YaleB [77] demonstrate the performance for face recognition. Caltech-101 [74] and Fifteen Scene [121] show the performance variation for object and scene categorization, respectively.

The approach as a function of the training data size for four different databases. In these experiments, we first divide each database into the test and training sets. A test set comprises randomly chosen samples from the database and it remains fixed throughout the experiments. In each experiment, the training data is generated by randomly selecting the samples from the training set. For the AR database [140], the test set is fixed to 5 samples per class. For Extended YaleB [77] we choose 24 samples per class for testing. A varied number of test samples per class are used for Caltech-101 [74] and Fifteen Scene Category [121]. For Caltech-101, this number varies between 1 and 770, whereas 100 to 300 test samples per class are used for evaluating the performance on Fifteen Scene Category database. For each database, the choice of the sizes of the test and training sets is mainly governed by the total number of samples per class available in the database.

From Fig. D.5, we can see that our approach is able to obtain nearly 90% accuracy for the faces and scene category databases with 10 training samples per class. By increasing the amount of the training data, the accuracy keeps improving. For Caltech-101, the accuracy is nearly 75% when 30 samples per class were used for training. However, 15 samples per class were already enough to obtain nearly 70% accuracy. Keeping in view these results we can say that our approach generally requires 10 to 15 samples per class for sufficient performance for a typical image classification task. However, larger amount of training data is more beneficial for our approach.
E.1 Graphical Representation

The graphical representation of the proposed model is given in Fig. E.1.

![Graphical representation](image)

Figure E.1: Graphical representation.

E.2 Joint Probability Distribution

According to the proposed model, the joint probability distribution over the data of the $c^{th}$ class can be expressed as:

$$
P(\{y^c_i\}, \{h^c_i\}, \Phi, \Psi, \{z^c_i\}, \{s^c_i\}, \{t^c_i\}, \{\pi^c_k\}, \lambda_y, \lambda_h) =$$

$$\prod_{i=1}^{|\mathcal{I}_c|} \mathcal{N}(y^c_i | \Phi(z^c_i \odot s^c_i), 1/\lambda_y, I_L) \ 	ext{Gam} \ (e_o, f_o) \ \mathcal{N}(h^c_i | \Psi(z^c_i \odot t^c_i), 1/\lambda_h, I_C) \ 	ext{Gam} \ (e_o, f_o)$$

$$\prod_{k=1}^{(|\mathcal{K}|)} \mathcal{N}(\varphi_k | 0, 1/\lambda_{\varphi}, I_L) \mathcal{N}(\psi_k | 0, 1/\lambda_{\psi}, I_C)$$

$$\prod_{i=1}^{(|\mathcal{I}_c|)} \prod_{k=1}^{(|\mathcal{K}|)} \text{Bernoulli} \ (z^c_{ik} | \pi_{k,a}) \ \text{Beta} \left( \frac{a_o}{K}, \frac{b_o(K-1)}{K} \right)$$

$$\prod_{k=1}^{(|\mathcal{K}|)} \mathcal{N}(s^c_i | 0, 1/\lambda_{s}, I_{[\mathcal{K}]}) \ 	ext{Gam} \ (\lambda^c_s | e_o, d_o) \mathcal{N}(t^c_i | 0, 1/\lambda_{t}, I_{[\mathcal{K}]}) \ 	ext{Gam} \ (\lambda^c_t | e_o, d_o).$$

E.3 Gibbs Sampling Equations

Except for $z^c_{ik}$, the derivation of Gibbs sampling equations for the parameters is similar to the one provided in Appendix B.1.2. Hence, we do not provide the details
E.3. Gibbs Sampling Equations

of all the sampling equations to avoid repetition. Below, we derive the sampling equation for $z_{ik}^c$.

**Sample $z_{ik}^c$:** Once the dictionary and the classifier have been sampled, we must sample $z_{ik}^c$ based on the updated dictionary and the classifier. The posterior probability distribution over $z_{ik}^c$ can be expressed as,

$$p(z_{ik}^c | -) \propto N(y_{i^c_{vk}} | \varphi_k(z_{ik}^c \cdot s_{ik}^c), \lambda_{y_0}^{-1} I_L) \cdot N(h_{i^c_{vk}} | \psi_k(z_{ik}^c \cdot t_{ik}^c), \lambda_{h_0}^{-1} I_C) \cdot \text{Bernoulli}(z_{ik}^c | \pi_{k_0}^c).$$

It is straightforward to show that based on the above mentioned posterior

$$p(z_{ik}^c = 1 | -) \propto \pi_{k_0}^c \cdot \exp\left(- \frac{(y_{i^c_{vk}} - \varphi_k s_{ik}^c) \cdot \lambda_{y_0} I_L (y_{i^c_{vk}} - \varphi_k s_{ik}^c)}{2}\right) \cdot \exp\left(- \frac{(h_{i^c_{vk}} - \psi_k t_{ik}^c) \cdot \lambda_{h_0} I_C (h_{i^c_{vk}} - \psi_k t_{ik}^c)}{2}\right) \cdot \text{Bernoulli}(z_{ik}^c | \pi_{k_0}^c).$$

Let $p_1 = \pi_{k_0}^c \cdot \xi_1 \cdot \xi_2 \cdot \xi_3 \cdot \xi_4$. We can derive an expression for $p(z_{ik}^c = 0 | -)$ in a similar fashion, that comes out to be:

$$p(z_{ik}^c = 0 | -) \propto (1 - \pi_{k_0}^c) \cdot \exp\left(- \frac{(y_{i^c_{vk}} - \varphi_k s_{ik}^c) \cdot \lambda_{y_0} I_L (y_{i^c_{vk}} - \varphi_k s_{ik}^c)}{2}\right) \cdot \exp\left(- \frac{(h_{i^c_{vk}} - \psi_k t_{ik}^c) \cdot \lambda_{h_0} I_C (h_{i^c_{vk}} - \psi_k t_{ik}^c)}{2}\right).$$

Let $p_o = (1 - \pi_{k_0}^c) \cdot \xi_1 \cdot \xi_3$. Using $p_1$ and $p_o$, $z_{ik}^c$ can be sampled from the following normalized Bernoulli distribution:

$$z_{ik}^c \sim \text{Bernoulli}\left( \frac{p_1}{p_1 + p_o} \right).$$

Simplifying further:

$$z_{ik}^c \sim \text{Bernoulli}\left( \frac{\pi_{k_0}^c \xi}{1 - \pi_{k_0}^c + \xi \pi_{k_0}^c} \right),$$

where, $\xi = \xi_2 \cdot \xi_4$. 

Bibliography


[70] M. Elad, Sparse and redundant representation: From theory to application in signal and image processing (Springer-Verlag, New York, NY, USA, 2010).


Bibliography

1/2 sparsity-constrained nonnegative matrix factorization’, IEEE Transac-

and recognition of action attributes’, in 2011 International Conf. on Computer

[166] I. Ramirez, P. Sprechmann, and G. Sapiro, ‘Classification and clustering via
dictionary learning with structured incoherence and shared features’, in IEEE
Conf. on Computer Vision and Pattern Recognition (2010), 3501–3508.

[167] C. E. Rasmussen and C. K. I. Williams, Gaussian processes for machine learn-

no. 4, 1232–1249.

[169] W. Ren, G. Li, D. Tu, and L. Jia, ‘Nonnegative matrix factorization with
regularizations’, IEEE Journal on Emerging and Selected Topics in Circuits

[170] R. Rigamonti, M. Brown, and V. Lepetit, Are sparse representations really
relevant for image classification?, in IEEE Conference on Computer Vision
and Pattern Recognition (2011).

[171] F. Rodriguez and G. Sapiro, ‘Sparse representation for image classi-
fication: Learning discriminative and reconstructive non-parametric

maximum average correlation height filter for action recognition’, in IEEE
Conf. on Computer Vision and Pattern Recognition (2008), 1–8.

for optimizing per-pixel endmember sets’, IEEE Transactions on Geoscience


