12-2013

# Bayesian Dictionary Learning for Single and Coupled Feature Spaces

Li He
*University of Tennessee - Knoxville*, lhe4@utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Li He entitled "Bayesian Dictionary Learning for Single and Coupled Feature Spaces." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Computer Engineering.

Hairong Qi, Major Professor

We have read this dissertation and recommend its acceptance:

Husheng Li, Jens Gregor, Russell Zaretzki

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

# Bayesian Dictionary Learning for Single and Coupled Feature Spaces

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Li He

December 2013

*To my parents*

# Acknowledgements

I would like to thank all the individuals who have inspired, encouraged, and advised me in the preparation of this dissertation.

First and foremost, I would like to thank my advisor, Dr. Hairong Qi. Her willingness to support my work and her guidance throughout my studies has allowed me to develop my skills as a researcher within a supportive team environment. Her openness and determination gave me tremendous encouragement during my research. I thank her for that opportunity.

I would also like to thank Dr. Russell Zaretzki, for pointing me the right direction in the research and enormous help in the algorithm development.

I would further like to thank the other members of my committee: Dr. Husheng Li and Dr. Jens Gregor. I greatly appreciate their time and input to this dissertation.

Within the AICIP Lab, I owe many thanks to my fellow graduate students. I enjoyed the many conversations and discussions that have had a great impact on my research and myself as a person. Wei Wang, Jiajia Luo, Shuangjiang Li, Rui Guo, Zhibo Wang, Dayu Yang, Sangwoo Moon, Mahmut Karakaya, Bryan Bodkin, thank you very much.

Last but not the least, I express my deepest appreciation to my parents, for their unconditional love, support and encouragement.

*All models are wrong, but some are useful. - George E.P.Box*

# Abstract

Over-complete bases offer the flexibility to represent much wider range of signals with more elementary basis atoms than signal dimension. The use of over-complete dictionaries for sparse representation has been a new trend recently and has increasingly become recognized as providing high performance for applications such as denoise, image super-resolution, inpaiting, compression, blind source separation and linear unmixing. This dissertation studies the dictionary learning for single or coupled feature spaces and its application in image restoration tasks. A Bayesian strategy using a beta process prior is applied to solve both problems.

Firstly, we illustrate how to generalize the existing beta process dictionary learning method (BP) to learn dictionary for single feature space. The advantage of this approach is that the number of dictionary atoms and their relative importance may be inferred non-parametrically.

Next, we propose a new beta process joint dictionary learning method (BP-JDL) for coupled feature spaces, where the learned dictionaries also reflect the relationship between the two spaces. Compared to previous couple feature spaces dictionary learning algorithms, our algorithm not only provides dictionaries that customized to each feature space, but also adds more consistent and accurate mapping between the two feature spaces. This is due to the unique property of the beta process model that the sparse representation can be decomposed to values and dictionary atom indicators. The proposed algorithm is able to learn sparse representations that correspond to the same dictionary atoms with the same sparsity but different values

in coupled feature spaces, thus bringing consistent and accurate mapping between coupled feature spaces.

Two applications, single image super-resolution and inverse halftoning, are chosen to evaluate the performance of the proposed Bayesian approach. In both cases, the Bayesian approach, either for single feature space or coupled feature spaces, outperforms state-of-the-art methods in comparative domains.

# Table of Contents


**1 Introduction** **1**

1.1 Dictionary learning in single feature space . . . . . 1

1.2 Dictionary learning in coupled feature spaces . . . . . 2

1.3 Motivation . . . . . 4

1.4 Contributions . . . . . 7

1.5 Dissertation Outline . . . . . 8

**2 Literature Review** **9**

2.1 Dictionary Learning in Single Feature Space . . . . . 9

2.1.1 Problem formulation . . . . . 9

2.1.2 Efficient $\ell^1$ . . . . . 10

2.1.3 Elastic Net Extension of Lasso . . . . . 11

2.1.4 FOCUSS-CNDL . . . . . 12

2.1.5 K-SVD . . . . . 12

2.1.6 Beta Process Dictionary Learning . . . . . 13

2.2 Dictionary Learning in Coupled Feature Space . . . . . 19

2.2.1 Two-step Dictionary Learning . . . . . 20

2.2.2 Semi-coupled Dictionary Learning . . . . . 21

2.2.3 Bilevel Sparse Coding . . . . . 22

2.3 Single Image Super-Resolution . . . . . 24

2.3.1 Sparse Representation based Single Image Super-Resolution . 24

2.3.2 Coupled dictionary learning in single feature space . . . . . 25
2.3.3 Nonlocal self-similarities . . . . . 26
2.4 Inverse Halftoning . . . . . 26
2.5 Image Quality Assessment . . . . . 28
2.5.1 SSIM . . . . . 31
2.5.2 VIF . . . . . 32
2.5.3 GSM . . . . . 34

**3 Beta Process Dictionary Learning for Single Feature Space 36**

**4 Beta Process Joint Dictionary Learning for Coupled Feature Spaces 40**
4.1 Learning Model . . . . . 40
4.2 Gibbs-sampling Inference . . . . . 44

**5 Application of Single Image Super-Resolution 48**
5.1 Beta Process Dictionary Learning for Single Feature Space . . . . . 48
5.1.1 Performance Metric . . . . . 50
5.1.2 Dictionary Learning Results . . . . . 52
5.1.3 Single Image Super-Resolution Results . . . . . 54
5.2 Beta Process Joint Dictionary Learning . . . . . 60
5.2.1 Experimental Design . . . . . 62
5.2.2 Dictionary Learning Results . . . . . 65
5.2.3 Single Image Super-Resolution Results . . . . . 66

**6 Application of Inverse Halftoning 73**
6.1 Beta Process Dictionary Learning for Single Feature Space . . . . . 75
6.2 Beta Process Joint Dictionary Learning . . . . . 80

**7 Conclusions and Future Work 85**
7.1 Summary . . . . . 85
7.2 Future Research . . . . . 86

7.2.1 Improvement of the BP-JDL . . . . . 86
7.2.2 Evaluation . . . . . 88
7.2.3 Other Applications . . . . . 88

**Bibliography** **89**

**Appendices** **101**

A Variational Inference of Beta Process Joint Dictionary Learning . . . 102
A.1 The VB-E Step . . . . . 102
A.2 The VB-M Step . . . . . 103
B Publications . . . . . 106

**Vita** **108**

# List of Tables


5.1 Comparison of dictionary learning results. For the BP, the third column is the dictionary size $K$ inferred. The initial $K$ is set to 1024 for all experiments. The fifth column is the dictionary learning time (hours). The sixth column (Sparsity) is the average number of dictionary atoms used for $100,000$ training samples. For K-SVD, the initial $T_0$ in Eq. 2.7 is set to 20. Results were produced on a Dell T3500 Workstation with 2.66G Intel Xeon X5550 CPU and 12GB of RAM running Ubuntu and Matlab V7.12.0. . . . . 53

5.2 Comparison of super-resolution results. For the BP, the second column is the dictionary size $K$ inferred. The initial $K$ is set to 1024 for all experiments. The super-resolution results of images in 8 categories are shown in the last 8 columns in averaged PSNR(dB) and SSIM. Same patch size ($7 \times 7$) is used for all three srSR methods. . . . . 56

5.3 Comparison of average super-resolution reconstruction time (seconds). $K$ is dictionary size. The patch size is $7 \times 7$. The SR time of the BP dictionaries is on average 34% and 26% shorter than the SR time of the efficient $\ell^1$ dictionaries and the K-SVD dictionaries, respectively. . 58

5.4 Comparison of factor of 2 magnification super-resolution results. . . . 67

5.5 Comparison of factor of 3 magnification super-resolution results. . . . 68

6.1 Comparison of inverse halftoning results. . . . . 79

# List of Figures


2.1 An example of the dictionary based single image super-resolution. . 25

2.2 An example of the dictionary based inverse halftoning. Although input halftoned image looks like a grayscale image, its a binary image. . . . 29

3.1 Graphical representation of the beta process model. $\mathbf{v}_i, i = 1, 2, \ldots, N$ are training samples and we assume $\mathbf{v}_i = \mathbf{D}(\mathbf{z}_i \circ \mathbf{s}_i) + \epsilon_i$. For the coefficients $(\mathbf{z}_i \circ \mathbf{s}_i)$, $\mathbf{z}_i$ is a binary vector $(z_{i1}, \ldots, z_{iK})$ that indicates which dictionary atoms are used by $\mathbf{v}_i$ and $\mathbf{s}_i$ is a vector $(s_{i1}, \ldots, s_{iK})$ of coefficient values. $\mathbf{d}_k$, $\mathbf{s}_i$ and $\epsilon_i$ are Gaussian distributed with variance $P^{-1}\mathbf{I}_P$, $\gamma_s^{-1}\mathbf{I}_K$ and $\sigma^2\mathbf{I}_P$, respectively. $z_{ik}$ is Bernoulli distributed with parameter $\pi_k$ and $\pi_k$ is Beta distributed with parameters $\frac{a}{K}$ and $\frac{b(K-1)}{K}$. 38

4.1 Graphical representation of the BP-JDL model for coupled feature spaces. $\mathbf{x}_i$ and $\mathbf{y}_i, i = 1, 2, \ldots, N$ are training samples for each feature space and we assume $\mathbf{x}_i = \mathbf{D}^{(x)}(\mathbf{z}_i \circ \mathbf{s}_i^{(x)}) + \epsilon_i^{(x)}$. For the coefficients $(\mathbf{z}_i \circ \mathbf{s}_i^{(x)})$, $\mathbf{z}_i$ is a binary vector $(z_{i1}, \ldots, z_{iK})$ that indicates which dictionary atoms are used by $\mathbf{x}_i$ and $\mathbf{s}_i^{(x)}$ is a vector $(s_{i1}^{(x)}, \ldots, s_{iK}^{(x)})$ of coefficient values. $\mathbf{d}_k^{(x)}$, $\mathbf{s}_i^{(x)}$ and $\epsilon_i^{(x)}$ are Gaussian distributed with variance $P_x^{-1}\mathbf{I}_{P_x}$, $\gamma_{s^{(x)}}^{-1}\mathbf{I}_K$ and $\gamma_{\epsilon^{(x)}}^{-1}\mathbf{I}_P$, respectively. Similar distribution is assumed for $\mathbf{d}_k^{(y)}$, $\mathbf{s}_i^{(y)}$ and $\epsilon_i^{(y)}$. $z_{ik}$ is Bernoulli distributed with parameter $\pi_k$ and $\pi_k$ is Beta distributed with parameters $\frac{a}{K}$ and $\frac{b(K-1)}{K}$. 44

5.1 Super-resolution results of dictionaries learned using different standard deviation of error vectors. The max value of $\sigma = 0.064$ is the standard deviation of normalized training samples. . . . . 49
5.2 80 test images for super-resolution. The images are divided into 8 categories, including car, natural, portrait, building, animal, flower, medical and CG. Each category has 10 test images. . . . . 51
5.3 BP dictionary learning with different initial $K$. . . . . 52
5.4 Factor of 2 magnification dictionaries ($\mathbf{D}_h$) learned by the BP, the efficient $\ell^1$ and the K-SVD, respectively. The dictionary trained by the BP contains 629 atoms. Dictionaries trained by the efficient $\ell^1$ and the K-SVD contain 1024 atoms. Each atom is a $7 \times 7$ size image patch and is normalized for display purpose. . . . . 54
5.5 Comparison of super-resolution images reconstructed using the Bicubic, the BP, the efficient $\ell^1$ and the K-SVD, respectively. (a) low-resolution input images. (b) original high-resolution images used to create the low-resolution images. The upper two rows show the factor of 2 magnification results. The lower two rows show the factor of 3 magnification results. Generally, the sparse representation based SR is better than the Bicubic interpolation. The BP dictionary produces the best SR image quality. . . . . 58
5.6 Super-resolution reconstruction PSNR of different size of high-res patch. The average PSNR of 8 categories are shown in individual sub-figures. . . . . 59
5.7 Super-resolution reconstruction SSIM of different size of high-res patch. The average SSIM of 8 categories are shown in individual sub-figures. 59
5.8 Test results of different overlap during super-resolution of the *Lena* image. The patch size is $7 \times 7$. . . . . 60

5.9 Super-resolution results of dictionaries learned using different noise variance ratios. We set the same noise variance ratio for both feature spaces. . . . . 61
5.10 (a) Non-stochastic patches and (b) Stochastic patches. Patch size is $7 \times 7$. . . . . 64
5.11 10 training images. . . . . 65
5.12 BP-JDL infers dictionary size non-parametrically. . . . . 66
5.13 BP-JDL learned mapping matrix $\log(\mathbf{M})$ for 771-size dictionary. . . . 67
5.14 Effect of the overlap parameter on PSNR and SSIM of test image Lion. 70
5.15 Visual comparison of factor of 2 super-resolution results. The upper row shows the SR results of the image Lena. The lower row shows the SR results of the image House. . . . . 71
5.16 Visual comparison of factor of 3 super-resolution results. The upper row shows the SR results of the image Lion. The lower row shows the SR results of the image Car. . . . . 72

6.1 Halftoned Lena image using Floyd-Steinberg. Although the halftoned image (b) looks like a grayscale image, its actually a binary image. . . 74
6.2 Learned size 506 coupled dictionary for inverse halftoning using BP. Dictionary atoms are normalized for display purpose. . . . . 77
6.3 BP learning sparsity over iterations. . . . . 78
6.4 (a)Noise histogram of (halftoned lena - lena), (b) Histogram of BP-JDL reconstructed lena, the pixel values are not between $[0, 1]$. . . . . 82
6.5 Inverse halftoning results comparison of Barbara. . . . . 83
6.6 Inverse halftoning results comparison of Lena. . . . . 84

# Chapter 1

# Introduction

## 1.1 Dictionary learning in single feature space

The use of over-complete dictionaries for sparse representation has been the subject of extensive research over the last decade. Research on signal processing (Mallat and Zhang, 1993) suggests that over-complete bases offer the flexibility to represent much wider range of signals with more elementary basis atoms than the signal dimension. In the field of early vision, the spatial receptive fields of simple cells have been characterized as being *localized, oriented* and *bandpass* (Olshausen and Fieldt, 1996), which cannot be captured in terms of linear, pairwise correlations such as principle components analysis (Hancock et al., 1991). (Olshausen and Fieldt, 1996, 1997) suggest that dictionary generated by sparse coding can capture the properties of receptive fields. They also suggests that image patches can be well represented as a sparse linear combination of elements from an appropriately chosen over-complete dictionary.

A straightforward way to obtain the dictionary for sparse representation is to sample image patch directly. However, this strategy will result in large dictionary and hence expensive computation. Therefore, to learn a compact dictionary is necessary to reduce the computational cost. There have been numerous methods proposed to

design such over-complete dictionaries, such as basis pursuit (Chen et al., 1998) or lasso (Tibshirani, 1996) or efficient $\ell^1$ (Lee et al., 2007), overcomplete ICA (Lewicki et al., 1998), RVM (Tipping, 2001), Method of Optimal Directions (MOD) (Engan et al., 1999), KSVD (Aharon et al., 2006), Energy-based model (Ranzato et al., 2006), elastic net (Zou and Hastie, 2005), FOCUSS based dictionary learning (Kreutz-Delgado and Rao, 2002; Murray and Kreutz-Delgado, 2007), online dictionary learning (Mairal et al., 2009a), and beta process dictionary learning (Paisley and Carin, 2009; Zhou et al., 2009). All these methods are able to generate the over-complete dictionary and sparse coefficients. Dictionaries learned by these methods yield sparse representations that have higher recovery accuracy than do with conventional representations, therefore attaining state-of-the-art performances on denoising, in-painting, image abstraction and super-resolution.

## 1.2 Dictionary learning in coupled feature spaces

In many signal processing problems, we have coupled feature spaces, e.g., the image patch space and sketch patch space for photo-sketch abstraction (Tang and Wang, 2003; Wang and Tang, 2009), the original and compressed signal spaces in compressive sensing (Yang et al., 2012a), and the high-resolution patch space and low-resolution patch space in patch-based image super-resolution (Yang et al., 2008), artistic rendering (Hertzmann et al., 2001; Efros and Freeman, 2001; Lin and Tang, 2005), multi-modal biometrics (Lei and Li, 2009; Sharma and Jacobs, 2011; Wang and Tang, 2009), inverse halftoning (Son, 2012; Mairal et al., 2012) and intrinsic image estimation (Jia et al., 2013). For the patch-based image super-resolution, many methods (Sun et al., 2003; Chang et al., 2004; Yang et al., 2008; Zeyde et al., 2010; Wang et al., 2012; Yang et al., 2012a) have been proposed trying to capture the concurrent prior between the low- and high-resolution patches using dictionary learning techniques. In these methods, a high-res patch is normally recovered using the high-res dictionary and sparse coefficients calculated using the

low-res feature patch and low-res feature dictionary. Therefore, we need to learn these two dictionaries in both high-res and low-res feature spaces. This is a typical dictionary learning problem in coupled feature spaces.

Many methods have been proposed to solve aforementioned dictionary learning problems in couple feature spaces such as patch-based matching (Hertzmann et al., 2001; Wang and Tang, 2009), coupled subspace learning (Lei and Li, 2009; Lin and Tang, 2005) and coupled dictionary learning (Yang et al., 2008). The intuitive method to learn dictionaries for coupled feature spaces is using single sparse coding model to learn the coupled dictionaries in concatenated spaces (Yang et al., 2008). Once the dictionaries are learned, we can use one dictionary to calculate the sparse coefficients and the other dictionary to recover the desired signal. However, dictionaries learned this way usually cannot capture the complex, spatial-variant and nonlinear relationship between the two feature spaces. In addition, because the sparse coefficients are shared between the two dictionaries, the algorithm normally finds it difficult to fit the dictionary and coefficients to both feature spaces. Therefore, a further learning model is necessary to adapt the dictionary learning algorithm to coupled feature spaces.

Several algorithms have been proposed to solve this problem (Zeyde et al., 2010; Wang et al., 2012; Yang et al., 2012a). Zeyde (Zeyde et al., 2010) proposed a two-step learning algorithm, where one dictionary is learned by KSVD (Aharon et al., 2006) and the other is generated via least-square. Zeyde used this approach for the single image super-resolution problem. Although this method largely decreases the computational cost because only one dictionary is learned and the dictionary is well-fitted in the low-res patch space, the same is not true in the high-res patch space. In addition, although the dictionaries are learned individually, same coefficients are still used for the two feature spaces, limiting the dictionaries from being customized to both spaces. A simultaneous dictionary learning algorithm is thus essential to balance the learning errors in both feature spaces.

The most recent approaches, also referred to as the semi-coupled approaches (Yang et al., 2012a; Wang et al., 2012), seek to improve the learning result by letting the dictionaries fit the two feature spaces better. Wang (Wang et al., 2012) proposed a semi-coupled training model to solve the problem where a mapping matrix is used to capture the relationship of the sparse representations between spaces. Although the learned dictionaries can better minimize the error in both spaces than those learned in concatenated spaces, the corresponding relationship of dictionaries in the two feature spaces are not captured during the learning process. Yang (Yang et al., 2012a) provided a bilevel sparse coding solution of the problem. Instead of solving the two optimization problems in two feature spaces together (Yang et al., 2010), the bilevel method moves one of the optimization problem to the regularization term of the other problem. Although the learned sparse representation of bilevel method has less learning errors, the same sparse coding is still required for both feature spaces. In addition, Yang's method also did not enforce the corresponding relationship between the learned dictionaries. These problems can be resolved by taking advantage of the beta process prior model.

## 1.3 Motivation

In this dissertation, we firstly consider the beta process (BP) for single feature space. Next, we refined BP to a new algorithm, beta process joint dictionary learning (BP-JDL), to better solve the problem of dictionary learning for coupled feature spaces.

Recent research on using non-parametric Bayesian approach (Griffiths and Ghahramani, 2005; Paisley and Carin, 2009) to learn an over-complete dictionary offers several advantages not found in earlier approaches and shows significant improvement in applications such as image denoising, inpainting and compressive sensing (Zhou et al., 2012). The advantage of using non-parametric Bayesian approach is the number of dictionary atoms and their relative importance may be inferred non-parametrically. In previous over-complete dictionary learning methods used in

application such as single image super-resolution (SISR) (Yang et al., 2008), the dictionary size is an unknown parameter and a large size dictionary is necessary to produce good super-resolution (SR) results based on the experience. In addition, in many applications, the desired sparsity level need to be manually set (Yang et al., 2010; Aharon et al., 2006). However, these two parameters are better to be inferred automatically. The Bayesian method may infer a smaller size dictionary non-parametrically and produce the same or better SR results. For example, for the factor of 2 magnification SR dictionary learning, the results show that Baysian method learned a 38.6% smaller dictionaries while produces better SR results. Since super-resolution using a smaller size dictionary needs less computational power, it may significantly affect the speed and energy consumption of super-resolution applications in resource-constrained environments. There has been recent interest in applying non-parametric Bayesian methods (Knowles and Ghahramani, 2007; Rai and Daumé III, 2008) to infer number of dictionary atoms, based on the observed data. Examples of recent research in this direction employs the Indian Buffet Process (IBP) (Griffiths and Ghahramani, 2005) and the beta process (BP) (Paisley and Carin, 2009). BP is more suitable for dictionary learning compared to IBP because it has more flexibility. In addition to previous mentioned advantages, BP also has the sparseness property that found in other dictionary learning algorithms, allowing dictionary size to tend to infinity while the training samples only use a small subset of dictionary atoms via the sparse coefficients.

**Beta process for single feature space.** In this article, the BP is considered for the single feature space dictionary learning problem. (Zhou et al., 2009) used a Gaussian distribution for the error vectors because the Gaussian noise is added to the signal in the denoise application. However, in many other single feature space dictionary learning problem, the error vectors are not necessarily Gaussian. In addition, the quality of the learned dictionary is sensitive to the model of error vectors, therefore the original model may not suitable for other applications. We revise

the model and use a pre-define parameter for the error vectors, therefore the revised model could be used for other single feature space dictionary learning applications.

**Beta process for coupled feature spaces.** Although BP provide a dictionary solution for single feature space, it may not be suitable to learn dictionaries in coupled feature spaces. Nevertheless, the truncated beta process allows the sparse coefficients to be expressed as an element-wise multiplication of a *binary* latent factor indicator and a *normal* coefficient value. We can take advantage of this property in the dictionary learning problem of coupled feature spaces by restraining the coefficients in coupled feature spaces to use the same dictionary atom indicator but different coefficient values.

Next, we propose a beta process joint dictionary learning algorithm for dictionary learning problems in coupled feature spaces. Compared to BP method used for single feature space, the new beta process model is customized for the problem of learning dictionaries in coupled feature spaces. Our model, together with Wang et al. (2012); Yang et al. (2012a), provides dictionary learning methods that customized to each feature space, however, our method adds more consistent and accurate mapping between the two feature spaces. This is due to the unique property of the beta process model Paisley and Carin (2009) that the sparse representations can be decomposed to values and dictionary atom indicators. We use the same beta process prior for dictionary atom indicators but different priors for values in two feature spaces. In this way, the BP-JDL is able to learn sparse representations that correspond to the *same* dictionary atoms with the *same* sparsity but *different* values in coupled feature spaces, thus bringing consistent and accurate mapping between coupled feature spaces. BP-JDL is able to learn the latent structure that customized to each feature space and provide a mapping function that reveals the complex relationship between the two feature spaces. In addition, BP-JDL inherits the advantage of BP: BP-JDL may also infer dictionary size non-parametrically and produce the same or better learning accuracies with much smaller dictionary size.

**Applications.** In order to compare the BP with other state-of-the-art single feature space dictionary learning methods, we tailor BP to the dictionary learning problem of single image super-resolution and inverse-halftoning. Experimental results show that dictionaries learned by BP produces the best super-resolution results compared to other two methods in the super-resolution application. In addition, BP outperformed the state-of-the-art inverse halftoning methods. Next, in order to compare BP-JDL with state-of-the-art coupled feature space dictionary learning methods, we tailor BP-JDL to the dictionary learning problem of the patch-based single image super-resolution as well as the inverse halftoning. Experimental results show that BP-JDL outperforms other methods in terms of both image quality for both applications.

## 1.4 Contributions

The primary goal of this research is to learn over-complete dictionaries for single or coupled feature spaces. To this end, the current contributions include:

- A revision of the beta process dictionary model (Paisley and Carin, 2009; Zhou et al., 2009) for single feature space. The quality of the dictionary is sensitive to the variance of the error vectors, thus the variance of the error vectors need to be pre-defined for many applications.
- A new beta process joint dictionary learning algorithm for coupled feature spaces. The learned sparse representation can minimize the recovery errors of each feature space while still capturing the relationship of the two spaces.
- An evaluation of revised beta process dictionary learning with applications of single image super-resolution and inverse halftoning.
- An evaluation of proposed beta process joint dictionary learning algorithm with application of single image super-resolution and inverse halftoning.

## 1.5 Dissertation Outline

The dissertation is organized as follows:

Chapter 2 provides a literature survey on state-of-the-art approaches on over-complete dictionary learning, image super-resolution, inverse halftoning and image quality assessment; Chapter 3 explains the generalization of beta process dictionary learning (Zhou et al., 2009) to single feature space dictionary learning problems (BP); Chapter 4 introduces the beta process joint dictionary learning for coupled feature spaces (BP-JDL); Chapter 5 shows the results of BP and BP-JDL in the application of single image super-resolution; Chapter 6 shows the results of BP and BP-JDL in the application of inverse halftoning; Finally, the dissertation is concluded with accomplished and future work in Chapter 7.

# Chapter 2

# Literature Review

In this chapter, we firstly describes several approaches used for over-complete dictionary learning in single feature space. We also review the beta process dictionary learning for single feature space and study the sensitive parameter. Next, we review several state-of-the-art over-complete dictionary learning algorithms for coupled feature spaces. For the applications of proposed dictionary learning algorithms, we describe the problem of single image super-resolution, inverse halftoning and how these image restoration problems can be formulated as dictionary learning problem. Finally, we review stat-of-the-art image quality measure methods and use these methods to measure the image quality image restoration results.

## 2.1 Dictionary Learning in Single Feature Space

### 2.1.1 Problem formulation

Given a set of training examples $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_N) \in \Re^{P \times N}$ in a single feature space, the sparse coding problem is to learn a dictionary $\mathbf{D} = (\mathbf{d}_1, \ldots, \mathbf{d}_K) \in \Re^{P \times K}$ that can sparsely represent the training examples with coefficients $\alpha \in \Re^{K \times N}$. $P$ is the

dimension of data and $K$ is the dictionary size. This problem is described as

$$\min \|\alpha\|_t \text{ s.t. } \|\mathbf{D}\alpha - \mathbf{V}\|_2^2 \leq \epsilon \tag{2.1}$$

where $t$ is a sparsity regularization. Ideally we want $t = 0$, then the problem become.

$$\min \|\alpha\|_0 \text{ s.t. } \|\mathbf{D}\alpha - \mathbf{V}\|_2^2 \leq \epsilon \tag{2.2}$$

However, this problem is non-convex and NP-hard. Many approximate solutions have been proposed to solve this problem.

### 2.1.2 Efficient $\ell^1$

Although the optimization problem of Eq. 2.2 is NP-hard in general, Donoho Donoho (2006) suggests that as long as the desired coefficients $\alpha$ are sufficiently sparse, they can be efficiently recovered by instead minimizing the $\ell^1$-norm, as follows:

$$\min \|\alpha\|_1 \text{ s.t. } \|\mathbf{D}\alpha - \mathbf{V}\|_2^2 \leq \epsilon \tag{2.3}$$

Using Lagrange multipliers, this equation can also be expressed as

$$(\mathbf{D}, \alpha) = \min_{\mathbf{D},\alpha} \frac{1}{2}\|\mathbf{V} - \mathbf{D}\alpha\|_2^2 + \lambda\|\alpha\|_1 \tag{2.4}$$

where the $\lambda$ balances sparsity of the solution and fidelity of the approximation to $\mathbf{X}$. The $\ell^1$ norm is to enforce the sparsity of $\alpha$. Eq. 2.4 is not convex in both $\mathbf{D}$ and $\alpha$, but it is convex in one of them with the other being fixed. There have been several algorithms proposed to solve this problem. (Yang et al., 2008) used the method proposed in (Lee et al., 2007) for dictionary learning of sparse coding based super-resolution (ScSR). The basic idea is to alternatively minimize Eq. 2.4 over $\alpha$ for a given dictionary $\mathbf{D}$, and then over $\mathbf{D}$ for a given $\alpha$, leading to a local minimum of the

overall objective function. In this method, the dictionary size need to be predefined before the learning step.

### 2.1.3 Elastic Net Extension of Lasso

The $\ell^1$ penalty in LASSO has several limitations (Zou and Hastie, 2005). For example, in the "large $p$, small $n$ problem" case, the LASSO selects at most $n$ variable before it saturates. Also if there is a group of highly correlated variables, the LASSO tends to select one variable from a group and ignore the others. To overcome these limitations, the elastic net adds a quadratic part to the penalty, which when used alone is ridge regression. The elastic-net formulation of dictionary learning problem can be expressed as

$$(\mathbf{D}, \alpha) = \min_{\mathbf{D},\alpha} \frac{1}{2}\|\mathbf{V} - \mathbf{D}\alpha\|_2^2 + \lambda_1\|\alpha\|_1 + \frac{\lambda_2}{2}\|\alpha\|_2^2 \tag{2.5}$$

where $\lambda_1$ and $\lambda_2$ are regularization parameter. When $\lambda_2 = 0$, this leads to LASSO. Currently, people choose elastic-net formulation over the LASSO is mainly for stability reasons. Using a parameter $\lambda_2 > 0$ makes the problem of Eq. 2.4 strongly convex and ensure its unique solution to be Lipschitz with respect to $V$ and $\mathbf{D}$ with a constant depending on $\lambda_2$. However, the stability is not necessarily an issue when learning a dictionary for a reconstruction task. Since this article is mainly consider the image restoration problem, we do not consider the stability issue.

Eq. 2.5 can be solved via online dictionary approach proposed in (Mairal et al., 2009a). (Mairal et al., 2012) applied this method to many applications, such as handwritten digits classification, inverse halftoning, digital art authentication and compressive sensing.

### 2.1.4 FOCUSS-CNDL

For a given dictionary $\mathbf{D}$, the focal underdertermined system solver (FOCUSS) was developed to solve Eq. 2.1 for $t \leq 1$ (Gorodnitsky et al., 1995; Rao et al., 1999). (Kreutz-Delgado et al., 2003) extend this algorithm for dictionary learning problem. Similar to the Efficient $\ell^1$ algorithm, the dictionary learning problem can be formulated as

$$(\mathbf{D}, \alpha) = \min_{\mathbf{D},\alpha} \frac{1}{2}\|\mathbf{V} - \mathbf{D}\alpha\|_2^2 + \lambda\|\alpha\|_t \tag{2.6}$$

where $t \leq 1$. Also similar to the Efficient $\ell^1$, this problem can be solved via alternative minimization. Firstly, the $\mathbf{D}$ is fixed and the $\alpha$ is updated using the FOCUSS algorithm. Next, the dictionary is re-estimated. Because $t \leq 1$, when the $\mathbf{D}$ is fixed, the problem of estimate $\alpha$ is non-convex and FOCUSS is used to provide the solution for the non-convex problem.

### 2.1.5 K-SVD

Another solution to Eq. 2.2 is called *K-SVD* (Aharon et al., 2006), in which a generalized K-Means clustering process is proposed. The algorithm used two steps (similar to the Efficient $\ell^1$) to learn sparse representation: In the first step, $\mathbf{D}$ is fixed and $\alpha$ is obtained via the Orthogonal Matching Pursuit (OMP) algorithm. This step is described as follows:

$$\min \|\mathbf{D}\alpha - \mathbf{V}\|_2^2 \text{ s.t. } \|\alpha\|_0 \leq T_0 \tag{2.7}$$

where $T_0$ is the predefined sparsity level. The number of dictionary atoms $K$ is predefined as well. In the second step, a Singular Value Decomposition (SVD) of the error is used to update $\mathbf{D}$. This approach is an approximation of the $\ell^0$-norm solution.

### 2.1.6 Beta Process Dictionary Learning

The beta process factorial analysis model is firstly proposed by (Paisley and Carin, 2009) for the latent factorial analysis problem, and later used by (Zhou et al., 2009) for the image de-noising and in-painting problem. We can treat dictionary $\mathbf{D}$ as factors and $\alpha$ as factor loadings, therefore the dictionary learning problem becomes a factor analysis problem, where the beta process (BP) can be employed as a prior for factor analysis.

#### Beta Process

The beta process, first introduced by (Hjort, 1990) for survival analysis, is an independent increments, or Lévy process (Miller, 2011).

**Definition 2.1.1.** *A Lévy process in $\mathbb{R}$ or $\mathbb{R}^+$, respectively, is a right-continuous function $Y$ from $[0, \infty)$ to $\mathbb{R}$ or $\mathbb{R}^+$ for which $Y_0 = 0$ a.s. and $Y$ has stationary, independent increments. Let $Y_t$ be the value of $Y$ at $t$ (Fristedt and Gray, 1997).*

Note that since Lévy processes have stationary, independent increments, they are infinitely divisible. For prior for latent feature models, we are only interested in the special case of Lévy processes in $\mathbb{R}$, which are non-decreasing functions also known as *subordinators*. Next, the beta process can be defined as follows (Paisley and Carin, 2009):

**Definition 2.1.2.** *Let $\mathcal{D}$ be a measurable space and $\mathcal{B}$ is its $\sigma$-algebra. Let $H_0$ be a continuous probability measure on $(\mathcal{D}, \mathcal{B})$ known as base measure and $a$ be a positive scalar as known as concentration parameter. Then for all disjoint, infinitesimal partitions, $\{\mathbf{d}_1, \cdots, \mathbf{d}_K\}$ of $\mathcal{D}$, the beta process is generated as follows*

$$H(\mathbf{d}_k) \sim Beta(aH_0(\mathbf{d}_k), a(1 - H_0(\mathbf{d}_k))) \tag{2.8}$$

*with $k \to \infty$ and $H_0(\mathbf{d}_k) \to 0$ for $k = 1, \cdots, K$. This process is denoted $H \sim BP(aH_0)$.*

In order to apply the beta process for the dictionary learning problem, we can also see the partition of $\mathcal{D}$, $\{\mathbf{d}_1, \cdots, \mathbf{d}_K\}$, as the partition for each dictionary atom $\mathbf{d}_k$.

Because of the convolution properties of beta random variables, the beta process does not satisfy the Kolmogorov consistency condition, and is therefore defined in the infinite limit (Billingsley, 1995). In general, $a$ can be a function of $\mathbf{d}$, but this is not commonly used in latent feature priors and is set to constant here. (Thibaux and Jordan, 2007) later showed how BP could be used as a nonparametric latent feature prior and latter mirrored by (**?**).

In order to better understand the definition above, we also construct the beta process from Lévy measure. The definition of Lévy measure is

**Definition 2.1.3.** *A measure $\nu$ on $\mathbb{R} \setminus \{0\}$ is called a Lévy measure if*

$$\int_{\mathbb{R}\setminus\{0\}} (y^2 \wedge 1)\nu(dy) < \infty \tag{2.9}$$

*A measure $\nu$ on $\mathbb{R}^+$ is called a Lévy measure if*

$$\int_{(0,\infty)} (y \wedge 1)\nu(dy) < \infty \tag{2.10}$$

*(Fristedt and Gray, 1997)*

This means that $\nu$ is a Lévy measure if for all $\epsilon$, there is finite mass more than $\epsilon$ away from zero. $\nu$ is allowed to have infinite mass near the origin, but Eq. 2.9 defines how fast $\nu$ is allowed to grow near the origin. We then define $\nu$ to be the Lévy measure for beta process defined on a space $\mathcal{D} \times [0, 1]$,

$$\nu(d\mathbf{d}, d\pi) = a\pi^{-1}(1-\pi)^{a-1}d\pi H_0(d\mathbf{d}) \tag{2.11}$$

where $\pi_k = H(\mathbf{d}_k)$ in this case is a probability measure. Here $H_0$ is known as base measure (instead of using the Lebesgue measure in Lévy process). Note by using

a more generic $\mathbf{d}_0$ (e.g., in dictionary learning the value of dictionary atom can be between $[-1, 1]$), the domain $\mathcal{D}$ can also be more general than $[0, \infty)$ and in case when $H_0$ is not a constant multiple of the Lebesgue measure, the name "completely random measure" is more appropriate than Lévy process. Note that since $\nu$ has the term $a\pi^{-1}(1-\pi)^{a-1}$, it is an infinite, improper beta measure in $\pi$ and therefore has infinite mass. By Champell's theorem, for any $\epsilon > 0$, there are only a finite number of points with $\pi$ greater than $\epsilon$. By the Lévy-Itō decomposition, BP is the results of integrating the $\pi$, we can represent BP as discrete measure (Miller, 2011):

$$H(\mathbf{d}) = \sum_{k=1}^{K} \pi_k \delta_{\mathbf{d}_k} \tag{2.12}$$

This is also known as a set function form. Like the Drichilet Process (Ferguson, 1973), means for drawing $H$ are not obvious. In case of general beta process, $\pi$ does not serve as a probability mass function on $\mathcal{D}$, but rather as part of a new measure on $\mathcal{D}$ that parameterizes a Bernoulli process defined as (Paisley and Carin, 2009):

**Definition 2.1.4.** *Let the column vector, $z_i$, be infinite and binary with the $k^{th}$ value, $z_{ik}$, generated by*

$$z_{ik} \sim Bernoulli(\pi_k) \tag{2.13}$$

The new measure, $X_i(\mathbf{d}) = \sum_k z_{ik}\delta_{\mathbf{d}_k}$, is then drawn from a Bernoulli process.

By arranging samples of the infinite-dimensional vector, $z_i$, in matrix form, $\mathbf{Z} = [\mathbf{z}_1, \cdots, \mathbf{z}_N]$, the beta process is seen to be a prior over infinite binary matrix, with each row in the matrix $\mathbf{Z}$ corresponding to a partition $\mathbf{d}_k$. Now we can see that any binary matrix with size $Z \times N$ will have positive probability under this prior, means we have defined a valid nonparametric prior.

The beta-Bernoulli process prior on $Z$ defined above tell us the distribution, but it does not tell us how to actually generate samples from this prior. One strategy is using the stick breaking process defined in (Miller, 2011) to generate samples for this prior. We will review another strategy below.

### Marginalized Beta Process and the Indian Buffet Process

Sampling $H$ from the infinite beta process is difficult, but a marginalized approach is derived by (Paisley and Carin, 2009) in the same manner as the corresponding Chinese restaurant process (Aldous, 1985), used for sampling from the Dirichlet process.

The beta process can be extend to take two scalar parameters, $a, b$, and the partition $\mathcal{D}$ into $K$ regions of equal measure, or $H_0(\mathbf{d}_k) = 1/K$ for $k = 1, \ldots, K$. We can then write the generative process in the form of Eq. 2.12 as

$$\begin{aligned} H(\mathbf{d}) &= \sum_{k=1} \pi_k \delta_{\mathbf{d}_k} \\ \pi_k &\sim Beta(a/K, b(K-1)/K) \end{aligned} \tag{2.14}$$

where $\mathbf{d} \in \{\mathbf{d}_1, \ldots, \mathbf{d}_K\}$. Marginalizing the vector $\pi$ and letting $K \to \infty$, the matrix $Z$ can be generated directly from the beta process prior as follows:

- 1. Initial $\mathbf{Z}$ to all zeros, set the first $c_1$ rows of $\mathbf{z}_1$ to 1, where $c_1 \sim Poisson(a/b)$. Sample the associated location, $\mathbf{d}_i, i = 1, \ldots, c_1$.
- 2. For observation $N$, sample $C_N \sim Poisson(\frac{a}{b+N-1})$ and define $C_N \equiv \sum_{i=1}^{N} c_i$. For rows $k = 1, \ldots, C_{N-1}$ of $\mathbf{z}_N$, sample

$$z_{Nk} \sim Bernoulli(\frac{n_{Nk}}{b+N-1}) \tag{2.15}$$

  where $n_{Nk} \equiv \sum_{i=1}^{N-1} z_{ik}$, the number of previous observation with a 1 at location $k$. Set indices $C_{N-1}+1$ to $C_N$ equal to 1 and sample associated locations.

If we define

$$H(\mathbf{d}) = \sum_{k=1}^{\infty} \frac{n_{Nk}}{b+N-1} \delta_{\mathbf{d}_k} \tag{2.16}$$

then $H \sim BP(a, b, H_0)$. As $N \to \infty$, the exchangeable columns of $\mathbf{Z}$ are drawn i.i.d. from a beta process. In the case where $b = 1$, the marginalized beta process is equivalent to Indian buffet process (Thibaux and Jordan, 2007).

From this definition, we see that the random variable $C_N$ has a Poisson distribution, $C_N \sim Poisson(\sum_{i=1}^{N} \frac{a}{b+i-1})$, which shows how binary matrix $Z$ grows with the sample size $N$. Furthermore, since $\sum_{i=1}^{N} \frac{a}{b+i-1} \to \infty$ as $N \to \infty$, we can deduce that the entire space of $\mathcal{D}$ will be explored as the number of samples grows to infinity.

We also can see that although (Paisley and Carin, 2009) showed that the $a, b$ parameters offer flexibility in tuning both the magnitude and shape of $\pi$, as $K$ become large, the $a, b$ parameters is non-informative and will not affect the shape of the distribution of $\pi$ since $\pi_k \sim Beta(a/K, b(K-1)/K)$.

### Finite Approximation to the Beta Process

A finite approximation of beta process can be made by simply setting $K$ to a large, but finite number. The finite representation may be written in set function form as

$$\begin{aligned} H &= \sum_{k=1}^{K} \pi_k \delta_{\mathbf{d}_k} \\ \pi_k \sim Beta(a/K, &b(K-1)/K), \quad \mathbf{d}_k \sim H_0 \end{aligned} \tag{2.17}$$

where $\delta_{\mathbf{d}_k}$ is a unit point mass at $\mathbf{d}_k$. $\pi_k$ represents a vector of $K$ probabilities, each associated with a respective dictionary atom $\mathbf{d}_k$. $H$ is composed by infinite number of $\mathbf{d}_k$ sampled from $H_0$ and is a valid measure when $K \to \infty$.

### Beta Process Dictionary Learning

Now we can show how to use beta process for dictionary learning as introduced in (Zhou et al., 2009). In order to use the beta process for factor analysis, the factor analysis problem can be expressed as(Paisley and Carin, 2009):

$$\mathbf{v}_i = \mathbf{D}\mathbf{z}_i + \epsilon \tag{2.18}$$

In order to represent which dictionary atoms each $\mathbf{v}_i$ uses, $N$ *binary* vectors $\mathbf{z}_i \in \{0,1\}^K, i = 1, \ldots, N$ are drawn from $H$ and the $k$th component of $\mathbf{z}_i$ is drawn from $z_{ik} \sim Bernoulli(\pi_k)$. These $N$ binary column vectors are used to constitute the matrix $\mathbf{Z} \in \{0,1\}^{K \times N}$, with the $i$th column corresponding to $\mathbf{z}_i$ and the $k$th row associated with atoms $\mathbf{d}_k$. Now, the matrix $\mathbf{Z}$ is modeled as $N$ draws from a Bernoulli process parameterized by a beta process.

However, for the dictionary learning problem, if we only use $\mathbf{z}_i$ as the only coefficient, it is highly restrictive as it imposes that the coefficients of the dictionary must be binary. To address this problem, (Paisley and Carin, 2009) added weights $\mathbf{s}_i \sim N(0, \gamma_s^{-1}\mathbf{I}_K)$ as part of the coefficients as well. $\mathbf{I}_K$ is an identity matrix and $\gamma_s^{-1}\mathbf{I}_K$ means we use the same variance $\gamma_s^{-1}$ for $(s_{i1}, \ldots, s_{iK})^T$. Now we have the coefficients $\alpha_i = \mathbf{z}_i \circ \mathbf{s}_i$, where $\circ$ represents element-wise multiplication of two vectors. In this way, the beta process could be used for dictionary learning problem and the it is formulated as

$$\begin{aligned} \mathbf{v}_i &= \mathbf{D}\alpha_i + \epsilon \\ \alpha_i &= \mathbf{z}_i \circ \mathbf{s}_i \end{aligned} \tag{2.19}$$

For the purpose of building a fully conjugate model, the dictionary atoms $\mathbf{d}_k$ are drawn from a multivariate zero-mean Gaussian ($H_0$) with variance $P^{-1}\mathbf{I}_P$ and the error vector $\epsilon$ are drawn from a zero-mean Gaussian with variance $\gamma_\epsilon^{-1}\mathbf{I}_P$. The use of $P^{-1}\mathbf{I}_P$ and $\gamma_\epsilon^{-1}\mathbf{I}_P$ means that we use the same variance $P^{-1}$ for $(d_{k1}, \ldots, d_{kP})^T$ and the same variance $\gamma_\epsilon^{-1}$ for $(\epsilon_1, \ldots, \epsilon_P)^T$. In addition, because the Inverse-gamma distribution is conjugate with the Gaussian distribution, $\gamma_s$ and $\gamma_\epsilon$ are drawn from

the Gamma distributions. The full model may be expressed as

$$\begin{aligned}
\mathbf{v}_i = \mathbf{D}\alpha_i + \epsilon, \quad & \alpha_i = \mathbf{z}_i \circ \mathbf{s}_i \\
\mathbf{D} = (\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K), \quad & \mathbf{d}_k \sim N(0, P^{-1}\mathbf{I}_P) \\
\mathbf{s}_i \sim N(0, \gamma_s^{-1}\mathbf{I}_K), \quad & \epsilon \sim N(0, \gamma_\epsilon^{-1}\mathbf{I}_P) \\
\gamma_s \sim \Gamma(c, d), \quad & \gamma_\epsilon \sim \Gamma(e, f) \\
\mathbf{z}_i \sim \prod_{k=1}^{K} Bernoulli(\pi_k), \quad & \pi_k \sim Beta(a/K, b(K-1)/K)
\end{aligned} \tag{2.20}$$

Elements in Eq. 4.3 are in the conjugate exponential family, and therefore the posterior inference may be implemented via Gibbs-sampling method with analytic update equations.

(Zhou et al., 2009) used this model for image denoise problem. However, this model may not suitable to other single feature space dictionary learning problems, because the Gaussian noise assumption on other applications may not be true. For instance, the distribution of recovery error vectors in the single image super-resolution dictionary learning problem is not Gaussian. Therefore, a modification of the model is necessary to let it applicable to other dictionary learning problems.

## 2.2 Dictionary Learning in Coupled Feature Space

Suppose we have two coupled feature spaces $\mathcal{Y} \in \mathbb{R}^{P_y}$ and $\mathcal{X} \in \mathbb{R}^{P_x}$, where the features are sparse in terms of certain dictionaries. There exists a mapping function $\mathcal{F} : \mathcal{Y} \to \mathcal{X}$ that relates features in $\mathcal{Y}$ to the corresponding features in $\mathcal{X}$. Therefore, the relation of the dictionaries and the observations and the relation of the two feature spaces can be described as

$$\begin{aligned}
\mathbf{x}_i &= \mathbf{D}^{(x)}\alpha_i^{(x)} + \epsilon_i^{(x)} \\
\mathbf{y}_i &= \mathbf{D}^{(y)}\alpha_i^{(y)} + \epsilon_i^{(y)}
\end{aligned} \tag{2.21}$$

where $\mathbf{x}_i, \mathbf{y}_i, i = 1, \ldots, N$ are training samples with dimensions $P_x$ and $P_y$, respectively. $\mathbf{D}^{(x)} = (\mathbf{d}_1^{(x)}, \mathbf{d}_2^{(x)}, \ldots, \mathbf{d}_K^{(x)})$ and $\mathbf{D}^{(y)} = (\mathbf{d}_1^{(y)}, \mathbf{d}_2^{(y)}, \ldots, \mathbf{d}_K^{(y)})$ are dictionaries we want to learn in each space and both dictionaries have $K$ atoms. $\alpha_i^{(x)}$ and $\alpha_i^{(y)}$ are coefficients of each dictionary. $\epsilon_i^{(x)}$ and $\epsilon_i^{(x)}$ are the recovery errors.

The intuitive method to learn dictionaries for coupled feature spaces is using single sparse coding model to learn the coupled dictionaries in concatenated spaces (Yang et al., 2008), where dictionary learning problems in Eq. 2.21 is converted to Eq. 2.2 by using $\mathbf{V} = [\mathbf{X}; \mathbf{Y}]$ and $\mathbf{D} = [\mathbf{D}^{(x)}; \mathbf{D}^{(y)}]$. Once the dictionaries are learned, we can use one dictionary to calculate the sparse coefficients and the other dictionary to recover the desired signal. However, dictionaries learned this way usually cannot capture the complex, spatial-variant and nonlinear relationship between the two feature spaces. Therefore, several algorithms have been proposed to solve this issue.

### 2.2.1 Two-step Dictionary Learning

In order to provide a better learning algorithm for coupled feature spaces and also accelerate the dictionary leanring speed, (Zeyde et al., 2010) proposed a two-step dictionary learning algorithm, where $\mathbf{D}^{(y)}$ and $\alpha$ is learned first and $\mathbf{D}^{(x)}$ is solved via least square using training samples $\mathbf{X}$ and learned sparse representation $\alpha$. The K-SVD algorithm is used to the first dictionary:

$$\min \|\mathbf{D}^{(y)}\alpha - \mathbf{V}\|_2^2 \text{ s.t. } \|\alpha\|_0 \leq T_0 \tag{2.22}$$

Next, the $\mathbf{D}^{(x)}$ is learned via least square:

$$\mathbf{D}^{(x)} = X\alpha^+ = X\alpha^T(\alpha\alpha^T)^{-1} \tag{2.23}$$

Because the sparse coding algorithm is only used to learn dictionary $\mathbf{D}^{(y)}$, the computation cost is largely reduced compared to the dictionary learning algorithm

in concatenated space. In addition, during the construction of the feature space $\mathcal{Y}$, a dimension reduction step is performed to further accelerate the learning speed.

Although the two-step learning algorithm has a faster learning speed, the relationship between two dictionaries is not reflected during the learning procedure. Moreover, recovery errors on the two feature spaces are not balanced, and the method still use the same sparse representation for both feature spaces. However, those constrains can be further relaxed via a more flexible algorithm.

### 2.2.2 Semi-coupled Dictionary Learning

Wang (Wang et al., 2012) proposed a semi-coupled dictionary learning (SCDL) algorithm for cross-style transfer, which normally require dictionaries in coupled feature spaces. SCDL relaxed the strong regularization of "same sparse representation" of the concatenated spaces dictionary learning algorithm, and also introduced a mapping function between the sparse coefficients. SCDL formulated the dictionary problem as below:

$$\begin{aligned}\min_{\mathbf{D}^{(x)},\mathbf{D}^{(y)},\mathbf{M}} & \|\mathbf{X}-\mathbf{D}^{(x)}\alpha^{(x)}\|_F^2+\|\mathbf{Y}-\mathbf{D}^{(y)}\alpha^{(y)}\|_F^2+\gamma\|\alpha^{(x)}-\mathbf{M}\alpha^{(y)}\|_F^2 \\ & +\lambda_x\|\alpha^{(x)}\|_1+\lambda_y\|\alpha^{(y)}\|_1+\lambda_M\|\mathbf{M}\|_F^2 \\ \mathbf{s.t.} & \|\mathbf{d}_i^{(x)}\|_{\ell_2}\leq 1, \|\mathbf{d}_i^{(y)}\|_{\ell_2}\leq 1\end{aligned} \tag{2.24}$$

where $\gamma$,$\lambda_x$,$\lambda_y$,$\lambda_M$ are regularization parameters to balance the terms in the objective function and $\mathbf{d}_i^{(x)}$ and $\mathbf{d}_i^{(y)}$ are the atoms of $\mathbf{D}^{(x)}$ and $\mathbf{D}^{(y)}$, respectively. The objective function is not jointly convex to $\mathbf{D}^{(x)}$, $\mathbf{D}^{(y)}$ and $\mathbf{M}$. However, it is convex w.r.t each of them if others are fixed. Therefore, an iterative algorithm could be used to alternatively optimize the variables.

From the objective function Eq. 2.24, we can see that although it uses different coefficients for individual feature space, during the learning procedure, the relationship correspondence between two dictionaries are not enforced.

Wang also introduce the multi-model for the coupled feature spaces, instead of using one pair of dictionaries. A clustering step is performed first using the Coupled Gaussian Mixture Model introduced by (Lin and Tang, 2005):

$$\max_{\mathbf{W},\mathbf{c}} \prod_{i=1}^{N} P(\mathbf{u}_i, \mathbf{v}_i | \mathbf{W}_{\mathbf{c}_i}) \tag{2.25}$$

where

$$\mathbf{c}_i = \mathbf{arg}\max_{k} P(\mathbf{u}_i, \mathbf{v}_i | \mathbf{W}_k) \tag{2.26}$$

and $\mathbf{W}_k$ indicates a coupled Gaussian model $\mathbf{u} \sim \mathcal{N}(\mathbf{w}_{u,k}, \Sigma_{u,k})$ and $\mathbf{v} \sim \mathcal{N}(\mathbf{w}_{v,k}, \Sigma_{v,k})$. $\mathbf{c}$ are model indices for samples. A model selection procedure is integrated by optimizing the following function:

$$\begin{aligned} &\max_{\mathbf{M},\mathbf{c}} \prod_{i=1}^{N} P(\alpha_i^{(x)}, \alpha_i^{(y)} | \mathbf{M}_{\mathbf{c}_i}) \\ =&\min_{\mathbf{M},\mathbf{c}} \sum_{i=1}^{N} \|\alpha_i^{(x)} - \mathbf{M}_{\mathbf{c}_i} \alpha_i^{(y)}\|_2 \end{aligned} \tag{2.27}$$

where $\mathbf{M}$ are mapping in each cluster. For the application of single image super-resolution, SCDL divided the training set to 32 clusters and learned 32 pairs of dictionary. When using the dictionaries for signal reconstruction, the multi-model definitely is more robust since it more over-complete (It has 32 times number of dictionary atoms than other methods). However, we are not sure the effectiveness of SCDL learned one pair dictionary compared to other algorithms. Also, the model selection algorithm require much more time than other methods since it uses a large dictionary.

### 2.2.3 Bilevel Sparse Coding

In order to provide an algorithm that customized for each feature space in coupled feature spaces dictionary learning problem, (Yang et al., 2012a) introduced a bilevel

sparse coding method, where the coupled sparse coding model is formulated as a generic bilevel optimization problem. Suppose the dictionary $\mathbf{D}^{(x)}$ is learned first using a standard sparse coding method to sparsely represent features in $\mathcal{X}$. The goal is to learn a "coupled" dictionary $\mathbf{D}^{(y)}$ over $\mathcal{Y}$, such that the sparse representation $\alpha$ of any $y \in \mathcal{Y}$ in terms of $\mathbf{D}^{(y)}$ can be used to recover its corresponding $x \in \mathcal{X}$ with dictionary $\mathbf{D}^{(x)}$ as $x^* = \mathbf{D}^{(x)}\alpha$. The optimization for $\mathbf{D}^{(y)}$ can be formulated as:

$$\begin{aligned} &\min_{\mathbf{D}^{(y)}} \sum_{i=1}^{N} \|\mathbf{D}^{(x)}\alpha_i^{(y)} - \mathbf{x}_i\|_2^2 \\ &\textbf{s.t.} \quad \alpha_i^{(y)} = \textbf{arg}\min_{\alpha} \|\alpha\|_1, \textbf{s.t.} \|\mathbf{y}_i - \mathbf{D}^{(y)}\alpha\|_2^2 \leq \epsilon, \forall i \\ &\quad \|\mathbf{D}_k^{(y)}\|_2 \leq 1, \forall k \end{aligned} \tag{2.28}$$

Although the dictionary learning in coupled feature spaces still use the same coefficients, dictionaries are learned alternatively rather than simultaneously, compared to the concatenated space algorithm. This is a bilevel optimization problem because there is an optimization problem in the constrain of main optimization problem. Being generically non-convex and non-differentiable, bilevel optimization programs are intrinsically difficult (Colson et al., 2007). However, Eq. 2.28 can be solved via first-order projected stochastic gradient descent.

Similar to SCDL method, bilevel sparse coding learn the dictionaries alternatively rather than simultaneously, compared to the concatenated space dictionary learning algorithm. In this way, the learned dictionaries can fit both space better. However, the same sparse representation is still used in the bilevel method. In addition, the corresponding relationship between the two dictionaries are not enforced during the learning for both SCDL and bilevel method.

## 2.3 Single Image Super-Resolution

### 2.3.1 Sparse Representation based Single Image Super-Resolution

*Super-resolution* is a technique that enhances the resolution of an image or multiple images of the same scene. Classic approaches (Tipping and Bishop, 2003; Farsiu et al., 2004) of super-resolution normally require multiple low-resolution images of the same scene to generate a super-resolution image. However, SR image reconstruction is generally a severely ill-posed problem because of the insufficient number of low-resolution images, ill-conditioned registration and unknown blurring operators. The recently studied single image super-resolution (SISR) problem attempts to enhance the resolution of a single image via offline learned patch-based dictionaries (Yang et al., 2008; Wang et al., 2010, 2012; Yang et al., 2012a,b; Lu et al., 2012). The low-resolution (low-res) image is down-sampled from a blurred high-resolution (high-res) image and often the blurring kernel is unknown.

$$\mathbf{L} = \downarrow B\mathbf{H} \tag{2.29}$$

where $\mathbf{L}$ is the observed low-resolution image, $\mathbf{H}$ is the high-resolution image. $\downarrow$ represents a downsampling operator and $B$ represents a blurring filter.

We use the framework proposed in (Yang et al., 2008) for application in this article. An example of the dictionary based image super-resolution is shown in Figure 2.1. First, a sparse representation of a patch $l$ in the low-resolution image $\mathbf{L}$ is found with the low-resolution dictionary $\mathbf{D}_y$ (two patches in $\mathbf{D}_y$ are found to constitute feature of $l$). Next, the *same* sparse representation is used with the high-resolution dictionary $\mathbf{D}_x$ to recover a patch $h$ in the high-resolution image $\mathbf{H}$. Because two dictionaries are jointly used for the low- and high-resolution image patches and the property of the sparse representation, the recovery of the high-resolution image from the low-resolution image is guaranteed.

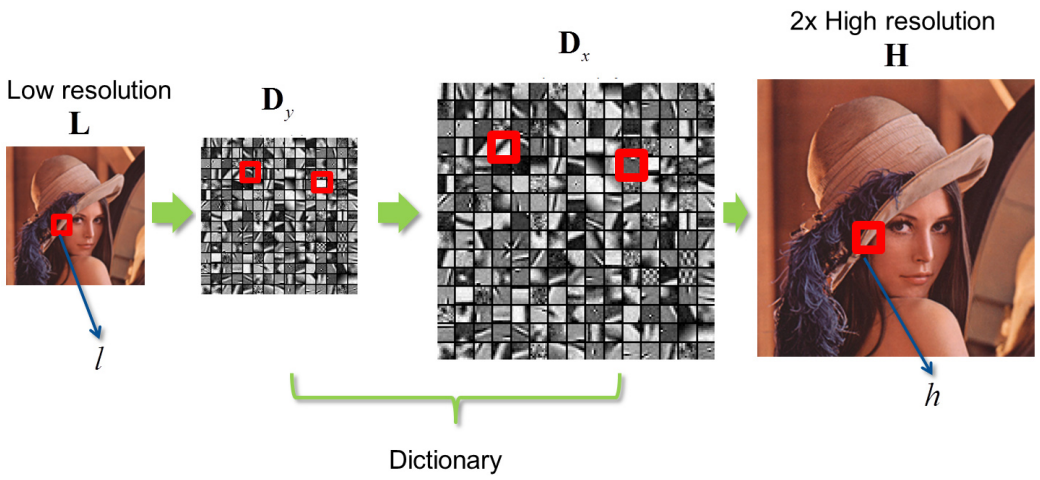



**Figure 2.1:** An example of the dictionary based single image super-resolution.

The dictionary learning problem for single image super-resolution can be formulated as an optimization problem:

$$\begin{aligned} \min \|\alpha\|_0 \text{ s.t. } & \|F\mathbf{D}_l\alpha - Fl\|_2^2 \leq \epsilon \\ & \|\mathbf{D}_h\alpha - h\|_2^2 \leq \epsilon \end{aligned} \tag{2.30}$$

where $\|\cdot\|_0$ is the number of non-zero elements in $\alpha$ known as the $\ell^0$-norm. $F$ are four (linear) feature extraction operators which are used to penalize visually salient high-frequency errors (Yang et al., 2008): $F_1 = [-1, 0, 1], F_2 = F_1^T, F_3 = [1, 0, -2, 0, 1], F_4 = F_3^T$. From Eq. 2.30, we can see this is a dictionary learning problem in coupled feature spaces.

### 2.3.2 Coupled dictionary learning in single feature space

The first approach that generated the state-of-the-art SISR result concatenates the two feature spaces together, thus converting the problem to dictionary learning in single feature space. In this way, the dictionaries are learned simultaneously via Eq. 2.2. The two feature spaces are constructed as:

$$\begin{aligned} \mathbf{x} &= h; \\ \mathbf{y} &= [F_1 l; F_2 l; F_3 l; F_4 l] \end{aligned} \tag{2.31}$$

After concatenates the two feature spaces, the constrained optimization of Eq. 2.30 can be reformulated as:

$$\min \|\alpha\|_0 \text{ s.t. } \|\mathbf{D}\alpha - \mathbf{v}\|_2^2 \leq \epsilon \tag{2.32}$$

where $\mathbf{D} = [\mathbf{D}_y; \mathbf{D}_x]$ and $\mathbf{v} = [\mathbf{y}; \mathbf{x}]$. Eq. 2.32 is the same as the Eq. 2.2, therefore can be solved by single feature space dictionary learning algorithms.

### 2.3.3 Nonlocal self-similarities

Recently many works have shown that the nonlocal redundancies existing in natural images are very useful for image restoration and a good combination of local sparsity and nonlocal redundancy can greatly enhance the performance of image reconstruction (Buades et al., 2005; Dabov et al., 2007a; Mairal et al., 2009b; Sun and Tappen, 2011; Wang et al., 2012). For a local patch $\mathbf{x}$, the nonlocal self-similarities searches for similar patches in the whole image, and predict this patch as:

$$\mathbf{x} = \sum_{m=1}^{M} b^m \mathbf{x}^m \tag{2.33}$$

where $\mathbf{x}^m$ is the $m^{th}$ most similar patch to $\mathbf{x}_i$ and $b^m$ is the nonlocal weight as defined in (Buades et al., 2005). We will use the nonlocal similarities in the super-resolution reconstruction step as a constraint of the recovery.

## 2.4 Inverse Halftoning

Digital halftoning has been widely used in digital printers, fax machines, and plasma display panel (PDP) TVs to create binary (halftoned) images with homogeneous black

and white dots from discrete images with 255 levels (TSUTOMU et al., 1999; Son and Choo, 2013). Many digital halftoning methods such as dithering, error diffusion, and direct binary search have been developed over the last several decades (Ho, 2004).

Inverse digital halftoning is the reverse of the digital halftoning proess, i.e., the reconstruction of a gray-level image from its halftoned version, which can correspond to the scanning process in scanner or copiers (Son and Choo, 2013). Without manually scanning the printed image, a gray-level image can be directly recovered from the saved halftoned image through an inverse halftoning method.

There are four major categories of inverse halftoning methods, namely, point spread function (PSF) based, look-up table (LUT) based, deconvolution based and sparse representation based methods. The simplest inverse halftoning method involves low-pass filtering of the input halftone image with a PSF that indicates the amount of direction of the blurring. It can remove most of the noise injected by the halftoned patterns, but it also removes the edge information. A method of using the maximum a-posterior (MAP) estimation has been developed to reconstruct both the smooth regions of the images and the discontinuities along the edges (Stevenson, 1997). Later, a fast inverse halftoning method of producing images with very good quality has been proposed based on a look-up table (LUT) wherein the relationship between a specific bit-pattern formed an ordering of 0 and 1 in a neighborhood and the corresponding average gray-level value given by the training image can be stored (Vaidyanathan, 2001). This method has been extend by (Chung and Wu, 2005) to edge-based LUT (E-LUT).

The deconvolution model of the error diffusion and its application to the inverse halftoning has been presented in (Kite et al., 2000), where a linear model of error diffusion is proposed and can be approximately represented by convolution of the original image and a PSF. Based on this deconvolution model, anisotropic deconvolution based on the regularized inverse-regularized Wiener inverse (RI-RWI) has been developed to allow nearly optimal edge adaptation. Later, (Neelamani et al., 2002) proposed to use wavelet (WInHD) for deconvolution model because the wavelet

transform preserve sharp edges. (Foi et al., 2004) further extend the deconvolution model to use a directional local polynomial approximation and the intersection of confidence intervals (LPA-ICI) algorithm, where the fine detail of image is better preserved. One drawback of the deconvolution based scheme is it assumes that the error diffusion kernel is known, where in many situation the kernel may be unknown.

Recently, the sparse representation based method have been developed for inverse halftoning task (Son, 2012; Mairal et al., 2012). It's a similar framework to the sparse representation based image super-resolution described in Section 2.3, where example based dictionaries are learned to explorer the implicit relationship between halftoned image and grayscale image. In this framework, a pair of overcomplete dictionaries are learned for halftoned and grayscale image simultaneously, and using the same sparse coding generate using halftoned dictionary the grayscale image can be reconstructed. (Son, 2012) used K-SVD to learn the dictionaries while (Mairal et al., 2012) used the LASSO to learn the dictionaries and generated better results.

In this article, we follow the framework proposed by (Mairal et al., 2012) and use beta process based dictionary algorithm to learn the dictionary. An example of the dictionary based image inverse halftoning is shown in Figure 2.2. First, a sparse representation of a patch $y$ in the halftoned image $\mathbf{Y}$ is found with the halftoned dictionary $\mathbf{D}_y$ (two patches in $\mathbf{D}_y$ are found to constitute feature of $y$). Next, the *same* sparse representation is used with the high-resolution dictionary $\mathbf{D}_x$ to recover a grayscale patch $x$ in the grayscale image $\mathbf{X}$.

## 2.5 Image Quality Assessment

Digital images are usually affected by a wide variety of distortions during acquisition and processing. Therefore, image quality assessment (IQA) is useful in many applications such as image acquisition, compressing, watermarking, restoration, enhancement and reproduction (Liu et al., 2012). The goal of IQA is to calculate the extent of quality degradation and is thus used to evaluate/compare the performance

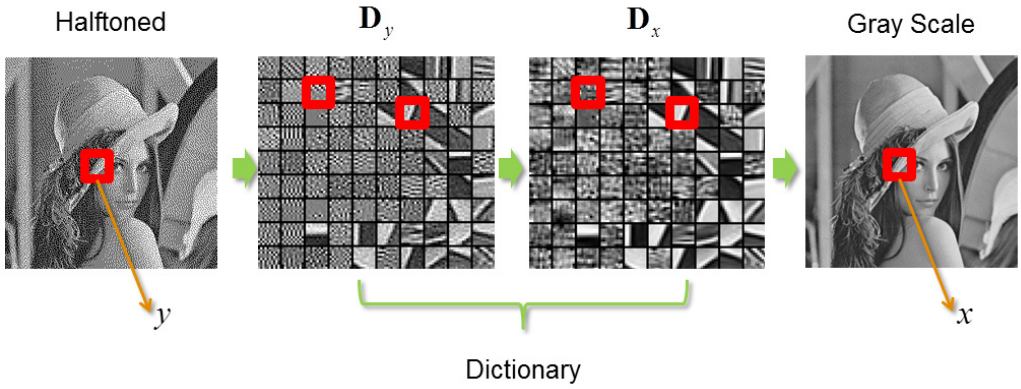



**Figure 2.2:** An example of the dictionary based inverse halftoning. Although input halftoned image looks like a grayscale image, its a binary image.

of processing systems and/or optimize the choice of parameters in processing. The human visual system (HVS) is the ultimate receiver of the majority of processed images, and evaluation based on subjective experiments is the most reliable way of IQA. However, subject evaluation is time consuming, laborious, expensive, and non-repeatable; as a result, it cannot be easily and routinely performed for many scenarios. These limitation have led to the development of *objective* IQA measures that can be easily embedded in image processing systems (Wang et al., 2004).

The simplest and most widely used IQA scheme is the mean squared error (MSE)/ peak signal-to-noise ratio (PSNR). However, the MSE/PSNR does not always agree with the subjective view results, particularly when distortion is not additive in nature (Wang et al., 2004). However, in this article, we will estimate the image quality via PSNR first as a baseline, then we also show measurement of other image quality measurement metrics for comparison as well.

In order to accurately and automatically evaluating the image quality in a manner that agrees with subject human judgments, regardless of the type of distortion corrupting the image, the content of the image, or the strength of the distortion, substantial research effort has been directed toward developing IQA schemes over the years, reviewed in (Wang et al., 2004). The well-known schemes proposed in recent years include structural similarity (SSIM) (Wang et al., 2004), visual information

fidelity (VIF) (Sheikh and Bovik, 2006), PSNR-HVS-M (N. Ponomarenko and Lukin, 2007), visual signal-to-noise ratio (VSNR) (Chandler and Hemami, 2007), most apparent distortion (MAD) (Larson and Chandler, 2010) and gradient similarity measurement (GSM) (Liu et al., 2012).

The SSIM (Wang et al., 2004) and VIF (Sheikh and Bovik, 2006) are based on high-level property of the images (e.g., structure information (Wang et al., 2004) or statistical information (Sheikh and Bovik, 2006)). They have demonstrated success for images containing suprathreshold distortions, and as a tradeoff, these schemes generally perform less well on images containing near-threshold distortions since such schemes do not adequately account for HVS' masking property (Larson and Chandler, 2010). The SSIM assumes that HVS is highly adapted for extracting structural information from a scene, and the SSIM is measured as the correlation between the two image blocks. The VIF views the IQA problem as an information fidelity problem, and the images are modeled using Gaussian scale mixtures to measure the amount of image information.

The PSNR-HVS-M (N. Ponomarenko and Lukin, 2007) use the PSNR in the discrete cosine transfer domain. The errors are weighted by the corresponding visibility threshold (which accounts for the masking effects of the HVS). However, (Wang et al., 2004) pointed out that there is no clear psychovisual evidence that the error visibility threshold based scheme is applicable to suprathreshold distortion.

The VSNR (Chandler and Hemami, 2007) deals with both detectability of distortion and structural degradation based on global precedence, a tradeoff for the performance on near-threshold and suprathreshold distortions is achieved. The MAD (Larson and Chandler, 2010) produces two quality scores, visibility-weighted error and the differences in log-Gabor subband statistics. Although it achieves good correlation with the human judgment, it has higher computational complexity.

(Liu et al., 2012) proposed a scheme based on edge/gradient similarity (GSM). The SSIM (Wang et al., 2004) is widely accepted due to its reasonably good evaluation accuracy (Gao et al., 2009), pixelwise quality measurement, and simple mathematical

formulation, which facilitates analysis and optimization. However, as pointed out in some existing works (Chen et al., 2006; Kim et al., 2010), it is less effective for badly blurred images since it underestimates the effects of edge damage and treats every region in an image equally. Edges are crucial for visual perception and play a major role in the recognition of image content (Ran and Farvardin, 1995; Ong et al., 2004). An example for the significance of edges comes form the fact that a mere sketch image can convey most information in the scene (Ran and Farvardin, 1995). Therefore, (Liu et al., 2012) explorer the edge/gradient similarity to evaluate the image quality. They demonstrated that gradient information captures both contrast and structure of images, allowing more emphasis on distortion around the edge regions. In addition, GSM integrate difference components (i.e. luminance and contrast-structure) of distortion.

In this article, we used MSE/PSNR, SSIM, VIF, GSM for image quality assessment. We review the detail of each approach below.

### 2.5.1 SSIM

The SSIM assumes that natural images are highly structured, and the HVS is sensitive to structural distortion. The structure information in an image is defined as those attributes that represent the structure of objects in the scene, independent of the average luminance and contrast (Wang et al., 2004).

The SSIM is calculated for each overlapped image block by using a pixel-by-pixel sliding window, and therefore it can provide the distortion/similarity map in the pixel domain. For any two image blocks $x$ and $y$, the SSIM models the similarity between them as three complementary components, namely, luminance similarity, contrast similarity, and structural similarity, formulated as below:

$$
\begin{aligned}
l(x,y) &= \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \\
c(x,y) &= \frac{\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \\
s(x,y) &= \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}
\end{aligned}
\tag{2.34}
$$

where $\mu_x, \mu_y, \sigma_x^2, \sigma_y^2$ and $\sigma_{xy}$ are the mean of $x$, the mean of $y$, the variance of $x$, the variance of $y$, and the covariance of $x$ and $y$, respectively;$C_1, C_2$ and $C3$ are claimed as small constants to avoid the denominator being zero.

The SSIM for the image blocks is given as

$$
SSIM(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s(x,y)]^{\gamma} \tag{2.35}
$$

where $\alpha$,$\beta$ and $\gamma$ are positive constraints used to adjust the relative importance of the three components. The higher the value of SSIM$(x,y)$ is, the more similar of image blocks $x$ and $y$ are. If $x$ and $y$ are the same, the SSIM value will be 1. The overall image quality score is determined using the mean of local SSIM. Similary scheme to SSIM include (Chen et al., 2006; Kim et al., 2010).

### 2.5.2 VIF

(Sheikh and Bovik, 2006) proposed visual information fidelity (VIF) to qualify the loss of image information to the distortion process and explore the relationship between image information and visual quality. It's a combination score of the amount of information shared between a reference and a distorted image and how much reference information can be extracted from the distorted image.

Firstly, the image formation model can be expressed as

$$
Y = \mathcal{G}X + n \tag{2.36}
$$

where $X$ is the reference image and $Y$ is distorted image. $\mathcal{G}$ is a deterministic scalar gain field and $n$ is a stationary additive zero-mean Gaussian noise with variance $\Sigma_n = \sigma_n^2 \mathbf{I}$. The HVS noise in wavelet domain can be modeled as

$$\begin{aligned} X &= \mathcal{C} + \epsilon \\ Y &= \mathcal{E} + \epsilon' \end{aligned} \tag{2.37}$$

where $\mathcal{C}$ denotes the random field (RF) from a subband in the reference image $X$ and $\mathcal{D}$ denotes the RF from the corresponding distorted image $Y$. $\epsilon$ and $\epsilon'$ are zero-mean uncorrelated multivariate Gaussian with the same covariance $\Sigma_\epsilon = \sigma_\epsilon \mathbf{I}$. Since $\mathcal{C}$ is a Gaussian scale mixtures, it can be expressed as a product of two independent RFs (Wainwright et al., 2001)

$$\mathcal{C} = \mathcal{S} \cdot \mathcal{U} \tag{2.38}$$

where $\mathcal{S}$ is an RF of positive scalars and $\mathcal{U}$ is a Gaussian vector RF with mean zero and covariance $\Sigma_U$. Next, we can use the mutual information $I(\mathcal{C}^N; X^N)$ to qualify the amount of information that can be extracted from the output of the HVS by the train when the image is being viewed ($\mathcal{C}^N = (\mathcal{C}_1, \ldots, \mathcal{C}_N)$ denote $N$ elements from $\mathcal{C}$). The information that could ideally be extracted by the brain from a particular subband $s^N$ in the reference image can be expressed as

$$I(\mathcal{C}^N; X^N | s^N) = \frac{1}{2} \sum_{i=1}^{N} \sum_{m=1}^{M} log_2(1 + \frac{s_i^2 \lambda_m}{\sigma_n^2}) \tag{2.39}$$

where $\lambda_m$ is an eigenvalue in a diagonal matrix $\Lambda$ and $\Sigma_U = \mathbf{Q}\Lambda\mathbf{Q}^T$ and $\mathbf{Q}$ is an orthonormal matrix. Also, the information that could ideally be extracted by the brain from a particular subband $s^N$ in the test image can be expressed as

$$I(\mathcal{C}^N; Y^N | s^N) = \frac{1}{2} \sum_{i=1}^{N} \sum_{m=1}^{M} log_2(1 + \frac{g_i^2 s_i^2 \lambda_m}{\sigma_n^2 + \sigma_\epsilon^2}) \tag{2.40}$$

Finally, the VIF is given by

$$VIF = \frac{\sum_j I(\mathcal{C}^{N,j}; Y^{N,j}|s^{N,j})}{\sum_j I(\mathcal{C}^{N,j}; X^{N,j}|s^{N,j})} \tag{2.41}$$

where $j$ is subbands of interest and $\mathcal{C}^{N,j}$ represent $N$ elements of the RF $\mathcal{C}_j$ that describes the coefficients from subband $j$, and so on. Therefore, VIF is the amount of information that brain could extracted from the test image *relative* to the amount of information that the brain could extract from the reference image. (Sheikh and Bovik, 2006) proved that the VIF is more consistence than SSIM for HVS in single-distortion as well as cross-distortion scenarios. However, the Gaussian noise assumption for the VIF may not true for many image distortion process.

### 2.5.3 GSM

(Liu et al., 2012) proposed a scheme that based on gradient similarity (GSM). The scheme contains two parts, gradient similarity and luminance distortion. The gradient similarity is defined as:

$$g(x, y) = \frac{2g_x g_y + C_4}{g_x^2 + g_y^2 + C_4} \tag{2.42}$$

where $g_x$ and $g_y$ are the gradient values for the central pixel of image blocks $x$ and $y$, respectively, and $C_4$ is the small constant to avoid the denominator being zero. $g(x, y)$'s value lies in $[0, 1]$. Gradient value $g_x$ (same for $g_y$ is calculated as the maximum weighted average of difference for the block:

$$g_x = \max_{k=1,2,3,4} mean2(|x \cdot M_k|) \tag{2.43}$$

where $M_k(k = 1, 2, 3, 4)$ are four kernels defined in (Liu et al., 2012). The $g(x, y)$ is able to measure both image contrast change and image structure change since the gradient value is a contrast-and-structure variant feature. Also, it less sensitive to

the case of higher masking contrast than of lower masking contrast, and is consistent with the contrast masking of the HVS for high masking contrast.

Next, the luminance distortion is measured by

$$e(x_i) = 1 - (\frac{x_i - y_i}{L})^2 \tag{2.44}$$

where $x_i$ and $y_i$ are the pixels at position $i$ in image blocks $x$ and $y$, respectively, and $L$ is the dynamic range of pixel values.

After calculate the gradient similarity and luminance similarity, a general form of integration to derive the overall quality indicator $q(x_i, y_i)$ for image pixel pair can be given as

$$q = (1 - W(g, e)) \cdot g + W(g, e) \cdot e \tag{2.45}$$

where $q$,$g$, and $e$ are the abbreviated forms of $q(x_i, y_i)$,$g(x_i, y_i)$ and $e(x_i, y_i)$, respectively. $W(g, e)$ is the weighting function used to adjust the relative importance of the two components.

The GSM could be used to gauge contrast and structural changes. In addition, for the dictionary learning based image restoration task such as super-resolution or inverse half-toning, the blocking effects could happen because we decompose the image to small patches. Compared to SSIM, GSM works better to measure the blocking effects of the image. Finally, in addition to the quality score, the GSM also can provide a quality map which may be used to view the quality distribution in the image.

# Chapter 3

# Beta Process Dictionary Learning for Single Feature Space

Zhou et al. (2009) used beta process dictionary learning in the image denoise application, where the noise in the image is synthetically added Gaussian noise. However, for the application such as image super-resolution, the distribution of error vectors, although close to is not exactly Gaussian. If we still use the inverse-Gamma distribution for the variance of error vectors, the Gaussian and inverse-Gamma model cannot fit the data well during the learning process. Therefore, we need to modify the model to adapt to the image SR application. We can still use the Gaussian model to model the error vectors, however, instead of Inverse-gamma distribution, a constant variance of error vectors is used to provide a lower bound for the variance of the error vectors. In other words, instead of using non-parametric Gaussian with hyper parameters, we use a parameter controlled Gaussian distribution to approximate the errors. In this way, we can learn the dictionary successfully. Moreover, the modified model could be used for other dictionary learning applications with non-Gaussian noise as well.

The beta process factorial analysis model is firstly proposed by (Paisley and Carin, 2009) for the latent factorial analysis problem, and later used by (Zhou et al., 2009)

for the image de-noising and in-painting problem. We can treat dictionary $\mathbf{D}$ as factors and $\alpha$ as factor loadings, therefore the dictionary learning problem becomes a factor analysis problem, where the beta process (BP) can be employed as a prior for factor analysis.

Following the general structure of beta process described in (Zhou et al., 2009). The modified beta process model used for the single feature space may be expressed as:

$$\begin{aligned} \mathbf{v}_i &= \mathbf{D}\alpha_i + \epsilon_i, \quad \alpha_i = \mathbf{z}_i \circ \mathbf{s}_i \\ \mathbf{D} &= (\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K), \quad \mathbf{d}_k \sim \mathcal{N}(0, P^{-1}\mathbf{I}_P) \\ \mathbf{s}_i &\sim \mathcal{N}(0, \gamma_s^{-1}\mathbf{I}_K), \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2\mathbf{I}_P) \\ \gamma_s &\sim \Gamma(c, d), \\ \mathbf{z}_i &\sim \prod_{k=1}^{K} Bernoulli(\pi_k), \quad \pi_k \sim Beta(a/K, b(K-1)/K) \end{aligned} \tag{3.1}$$

where $\mathbf{v}_i, i = 1, \ldots, N$ are training samples. Although this model is similar to Zhou et al. (2009), the variance of error vectors are set as constant ($\sigma^2\mathbf{I}_P$) instead of Inverse-gamma distributed. A graphical representation of this model is shown in Figure 3.1. In order to represent which dictionary atoms each $\mathbf{v}_i$ used, $N$ *binary* vectors $\mathbf{z}_i \in \{0,1\}^K, i = 1, \ldots, N$ are drawn from $H$ and the $k$th component of $\mathbf{z}_i$ is drawn from $z_{ik} \sim Bernoulli(\pi_k)$. These $N$ binary column vectors are used to constitute a matrix $\mathbf{Z} \in \{0,1\}^{K \times N}$, with the $i$th column corresponding to $\mathbf{z}_i$ and the $k$th row associated with atoms $\mathbf{d}_k$. In addition, weights $\mathbf{s}_i \sim N(0, \gamma_s^{-1}\mathbf{I}_K)$ are drawn as part of the coefficients as well. $\mathbf{I}_K$ is an identity matrix indicate that we use the same $\gamma_s^{-1}$ for all $(s_{i1} \ldots s_{iK})$. The coefficients $\alpha_i = \mathbf{z}_i \circ \mathbf{s}_i$, where $\circ$ represents element-wise multiplication of two vectors.

For the purpose of building a fully conjugate model, the dictionary atoms $\mathbf{d}_k$ are drawn from a multivariate zero-mean Gaussian ($H_0$) with variance $P^{-1}\mathbf{I}_P$ and the error vectors $\epsilon_i$ are drawn from a zero-mean Gaussian with variance $\sigma^2\mathbf{I}_P$. In addition,

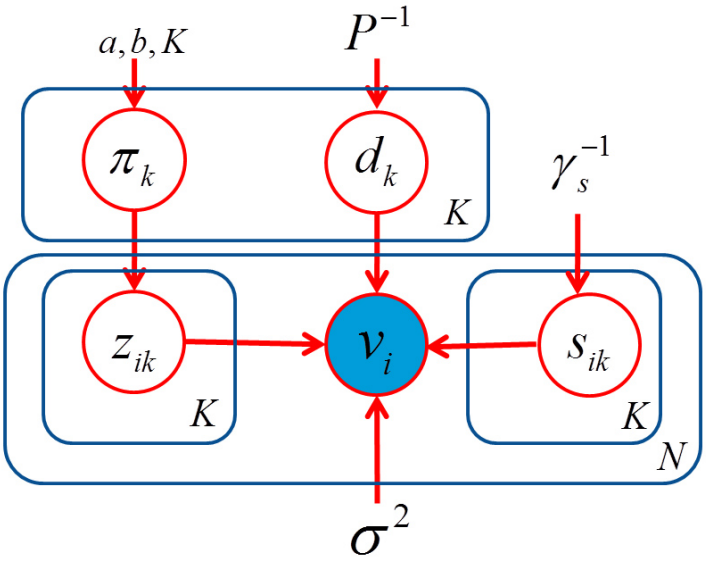



**Figure 3.1:** Graphical representation of the beta process model. $\mathbf{v}_i, i = 1, 2, \ldots, N$ are training samples and we assume $\mathbf{v}_i = \mathbf{D}(\mathbf{z}_i \circ \mathbf{s}_i) + \epsilon_i$. For the coefficients $(\mathbf{z}_i \circ \mathbf{s}_i)$, $\mathbf{z}_i$ is a binary vector $(z_{i1}, \ldots, z_{iK})$ that indicates which dictionary atoms are used by $\mathbf{v}_i$ and $\mathbf{s}_i$ is a vector $(s_{i1}, \ldots, s_{iK})$ of coefficient values. $\mathbf{d}_k$, $\mathbf{s}_i$ and $\epsilon_i$ are Gaussian distributed with variance $P^{-1}\mathbf{I}_P$, $\gamma_s^{-1}\mathbf{I}_K$ and $\sigma^2\mathbf{I}_P$, respectively. $z_{ik}$ is Bernoulli distributed with parameter $\pi_k$ and $\pi_k$ is Beta distributed with parameters $\frac{a}{K}$ and $\frac{b(K-1)}{K}$.

because the Inverse-gamma distribution is conjugate with the Gaussian distribution, $\gamma_s$ is drawn from the Gamma distributions. The Non-informative Gamma hyper-prior is placed on $\gamma_s$ (We initialize $c = d = 10^{-6}$). In this model, the expected number of factors (sparsity level) present in a training sample $\mathbf{v}_i$ as $K \to \infty$ is drawn from $Poisson(a/b)$. We set $a = b = 1$, but one may change values of $a$ and $b$. However, Zhou et al. (2009) proved the sparsity level is not sensitive to different values of $a$ and $b$ and is intrinsic to the data.

Elements in Eq. 3.1 are in the conjugate exponential family, and therefore the posterior inference may be implemented via a variational Bayesian or Gibbs-sampling method with analytic update equations. (Paisley and Carin, 2009) proposed an variational inference of the beta process and it converges around 10 iterations. However, each iteration may take long time and the dictionary learning time is as the same as the Gibbs sampling. Therefor in this paper, Gibbs-sampling is implemented.

For the initialize of the dictionary, we can either initial the dictionary values randomly or utilizing the SVD results of the input data samples. Experiment results show that either initialization approach can produce the dictionary successfully. In addition, we randomly initialize the coefficient values. For the binary matrix $\mathbf{Z}$, we initialize it to all zeros.

In the Gibbs-sampling process, after burnin samples, we exam that if each dictionary atom is used for the data at each iteration (check if $\sum_{i=1}^{N} z_{ik} = 0$). If the dictionary atom is not used, we delete the dictionary atom. If we start with a relatively large $K$, the $K$ will reduce during the Gibbs-sampling process. In this way, we can infer the appropriate dictionary size non-parametrically.

# Chapter 4

# Beta Process Joint Dictionary Learning for Coupled Feature Spaces

## 4.1 Learning Model

Suppose we have two coupled feature spaces $\mathcal{Y} \in \mathbb{R}^{P_y}$ and $\mathcal{X} \in \mathbb{R}^{P_x}$, where the features are sparse in terms of certain dictionaries. There exists a mapping function $\mathcal{F} : \mathcal{Y} \to \mathcal{X}$ that relates features in $\mathcal{Y}$ to the corresponding features in $\mathcal{X}$. Therefore, the relation of the dictionaries and the observations and the relation of the two feature spaces can be described as

$$\begin{aligned} \mathbf{x}_i &= \mathbf{D}^{(x)}\alpha_i^{(x)} + \epsilon_i^{(x)} \\ \mathbf{y}_i &= \mathbf{D}^{(y)}\alpha_i^{(y)} + \epsilon_i^{(y)} \\ \mathbf{M}\alpha_i^{(y)} &= \alpha_i^{(x)} \end{aligned} \tag{4.1}$$

where $\mathbf{x}_i, \mathbf{y}_i, i = 1, \ldots, N$ are training samples with dimensions $P_x$ and $P_y$, respectively. $\mathbf{D}^{(x)} = (\mathbf{d}_1^{(x)}, \mathbf{d}_2^{(x)}, \ldots, \mathbf{d}_K^{(x)})$ and $\mathbf{D}^{(y)} = (\mathbf{d}_1^{(y)}, \mathbf{d}_2^{(y)}, \ldots, \mathbf{d}_K^{(y)})$ are dictionaries learned in each space and both dictionaries have $K$ atoms. $\alpha_i^{(x)}$ and

$\alpha_i^{(y)}$ are coefficients of each dictionary. $\epsilon_i^{(x)}$ and $\epsilon_i^{(x)}$ are the recovery errors. $M$ is a mapping matrix from sparse coding of $\mathbf{y}_i$ to $\mathbf{x}_i$. In order to learn two dictionaries at the same time, previous algorithms (Yang et al., 2010, 2012a) use the same coefficients for both dictionaries, i.e., $\alpha_i^{(x)} = \alpha_i^{(y)}$. In this way, one might concatenate two feature spaces and convert the dictionary learning problem of coupled feature spaces to the dictionary learning problem of single feature space. However, allowing different coefficients in two feature spaces provides a better fitting of learning and the learned dictionaries are more customized to individual feature space. Beta process factor analysis (Paisley and Carin, 2009) allows the decomposition of the coefficients to the element multiplication of dictionary atom indicators and coefficient values, providing the much needed flexibility to fit each feature space better while still maintaining the correspondence between the two dictionaries.

We develop a new beta process based on (Zhou et al., 2012) to tackle the dictionary learning problem in coupled feature spaces. The new two-parameter beta process with parameters $a, b > 0$ and base measure $H_0$, is represented as $BP(a, b, H_0)$ and may be written in set function form as

$$\begin{aligned} H &= \sum_{k=1}^{K} \pi_k \delta_{\mathbf{d}_k^{(x)}} = \sum_{k=1}^{K} \pi_k \delta_{\mathbf{d}_k^{(y)}} \\ \pi_k &\sim Beta(a/K, b(K-1)/K), \quad \mathbf{d}_k^{(x)}, \mathbf{d}_k^{(y)} \sim H_0 \end{aligned} \tag{4.2}$$

where $\delta_{\mathbf{d}_k^{(x)}}$ and $\delta_{\mathbf{d}_k^{(y)}}$ are unit point mass at $\mathbf{d}_k^{(x)}$ and $\mathbf{d}_k^{(y)}$. We use a single beta process prior and the same dictionary atom indicator to connect the two feature spaces. In the beta process definition 2.1.2, the beta process is associate with one measurable space $\mathcal{D}$, and now we extend the beta process to associate with two measurable spaces $\mathcal{X}$ and $\mathcal{Y}$. $\mathbf{d}_k^{(x)}, k = 1, \ldots, K$ are partitions of $\mathcal{X}$ and $\mathbf{d}_k^{(y)}$ are partitions of $\mathcal{Y}$. $\pi_k$ represents a vector of $K$ probabilities, each associated with the respective atom $\mathbf{d}_k^{(x)}$ and the corresponding $\mathbf{d}_k^{()}$. Compared to the beta process dictionary learning model, the $\pi_k$ in the new model is associate with two partitions (one in $\mathcal{X}$ and one in $\mathcal{Y}$)

instead of one. $H$ is composed by infinite number of $\mathbf{d}_k^{(y)}$ (as well as corresponding $\mathbf{d}_k^{(x)}$) sampled from $H_0$ and is a valid measure when $K \to \infty$. Similar to beta process dictionary learning model in single feature space, a finite approximation of $H$ can be made by simply setting $K$ to a large, but finite number.

Following the general structure of beta process described in (Zhou et al., 2012), the beta process joint dictionary learning model for the coupled feature spaces may be expressed as

$$\begin{aligned}
\mathbf{x}_i = \mathbf{D}^{(x)}\alpha_i^{(x)} + \epsilon_i^{(x)}, \quad & \mathbf{y}_i = \mathbf{D}^{(y)}\alpha_i^{(y)} + \epsilon_i^{(y)} \\
\alpha_i^{(x)} = \mathbf{z}_i \circ \mathbf{s}_i^{(x)}, \quad & \alpha_i^{(y)} = \mathbf{z}_i \circ \mathbf{s}_i^{(y)} \\
\mathbf{d}_k^{(x)} \sim \mathcal{N}(0, P_x^{-1}\mathbf{I}_{P_x}), \quad & \mathbf{d}_k^{(y)} \sim \mathcal{N}(0, P_y^{-1}\mathbf{I}_{P_y}) \\
\mathbf{s}_i^{(x)} \sim \mathcal{N}(0, \gamma_{s^{(x)}}^{-1}\mathbf{I}_K), \quad & \mathbf{s}_i^{(y)} \sim \mathcal{N}(0, \gamma_{s^{(y)}}^{-1}\mathbf{I}_K) \\
\mathbf{z}_i \sim \prod_{k=1}^{K} Bernoulli(\pi_k), \quad & \pi_k \sim Beta(a/K, b(K-1)/K) \\
\epsilon_i^{(x)} \sim \mathcal{N}(0, \gamma_{\epsilon^{(x)}}^{-1}\mathbf{I}_{P_x}), \quad & \epsilon_i^{(y)} \sim \mathcal{N}(0, \gamma_{\epsilon^{(y)}}^{-1}\mathbf{I}_{P_y}) \\
\gamma_{s^{(x)}}, \gamma_{s^{(y)}} \sim \Gamma(c, d), \quad & \gamma_{\epsilon^{(x)}}, \gamma_{\epsilon^{(y)}} \sim \Gamma(e, f)
\end{aligned} \tag{4.3}$$

In order to constrain that $\mathbf{x}_i$ uses the same corresponding dictionary atom as that used by $\mathbf{y}_i$, we choose the same dictionary atom indicator $\mathbf{z}_i$ for both $\mathbf{d}_k^{(x)}$ and $\mathbf{d}_k^{(y)}$. At the same time, in order to provide different coefficient values, weights $\mathbf{s}_i^{(x)}$ and $\mathbf{s}_i^{(y)}$ are drawn from different distributions, as part of the coefficients. Finally we have the coefficients $\alpha_i^{(x)} = \mathbf{z}_i \circ \mathbf{s}_i^{(x)}$ and $\alpha_i^{(y)} = \mathbf{z}_i \circ \mathbf{s}_i^{(y)}$. Because $\alpha^{(y)}$ and $\alpha^{(x)}$ use the same dictionary atom indicator $\mathbf{z}_i$, they have the same number of non-zero elements and the corresponding relationship of dictionary atoms in the two feature spaces are enforced during the learning process.

Same to beta process dictionary learning model, $N$ *binary* vectors $\mathbf{z}_i \in \{0,1\}^K, i = 1, \ldots, N$ are drawn from $H$ and the $k$th component of $\mathbf{z}_i$ is drawn from $z_{ik} \sim Bernoulli(\pi_k)$. These $N$ binary column vectors are used to constitute the dictionary atom indicator matrix $\mathbf{Z} \in \{0,1\}^{K \times N}$, with the $i$th column corresponding to $\mathbf{z}_i$ and

the $k$th row associated with both $\mathbf{d}_k^{(x)}$ and $\mathbf{d}_k^{(y)}$. Compared to the beta process model for single feature space, now each $z_{ik}$ is associated with a partition in feature space $\mathcal{X}$ and a corresponding partition in feature space $\mathcal{Y}$.

Next, weights $\mathbf{s}_i^{(x)} \sim N(0, \gamma_{s^{(x)}}^{-1}\mathbf{I}_K)$ and $\mathbf{s}_i^{(y)} \sim N(0, \gamma_{s^{(y)}}^{-1}\mathbf{I}_K)$ are drawn as part of the coefficients. $\mathbf{I}_K$ is an identity matrix indicating that we use the same $\gamma_{s^{(x)}}^{-1}$ and $\gamma_{s^{(y)}}^{-1}$ for all $(s_{i1}^{(x)} \dots s_{iK}^{(x)})$ and $(s_{i1}^{(y)} \dots s_{iK}^{(y)})$. For the BP-JDL, we have different coefficients values for feature pair $\mathbf{x}_i$ and $\mathbf{y}_i$ while the feature pair use the same coefficient value if we concatenate the feature pair and use beta process model of single feature space for learning.

Similar to the beta process model we mentioned in the previous Chapter, the dictionary atoms $\mathbf{d}^{(\mathbf{x})}{}_k$ are drawn from a multivariate zero-mean Gaussian ($H_0$) with variance $P_x^{-1}\mathbf{I}_{P_x}$ and the error vectors $\epsilon^{(\mathbf{x})}{}_i$ are drawn from a zero-mean Gaussian with variance $\gamma_{\epsilon^{(x)}}^{-1}\mathbf{I}_P$. Next, $\gamma_{s^{(x)}}$ are drawn from the Gamma distributions. The non-informative Gamma hyper-prior is placed on $\gamma_{s^{(x)}}$ and $\gamma_{\epsilon^{(x)}}$, where we normally initialize $c = d = e = f = 10^{-6}$. We also apply the same distribution to $\mathbf{d}_k^{(y)}$, $\epsilon_i^{(y)}$, $\gamma_{s^{(y)}}$ and $\gamma_{\epsilon^{(y)}}$. In this model, the expected sparsity level in a training sample $\mathbf{x}_i$ or $\mathbf{y}_i$ as $K \to \infty$ is drawn from $Poisson(a/b)$. The sparsity level of the representation, is the influenced by the parameter $a$ and $b$. Examing the posterior of $p(\pi_k|-)$ in Section 4.2, conditioned on all other parameters, we find that most setting of $a$ and $b$ tend to be non-informative. Therefore, the average sparsity level of the coefficients is inferred by the data example itself. Each data example $x_i$ and $y_i$, has its own unique sparse representation based on the posterior, which renders much more flexibility than enforcing the same sparsity level of each example. We set $a = b = 1$. Finally, after we learned $\alpha^{(y)}$ and $\alpha^{(x)}$, the mapping matrix $\mathbf{M}$ can be calculated via the least square:

$$\mathbf{M} = [(\alpha^{(y)}\alpha^{(y)^T})^{-1}\alpha^{(y)}\alpha^{(x)^T}]^T \tag{4.4}$$

In the BP-JDL learning process, the two coefficients $\alpha_i^{(x)}$ and $\alpha_i^{(y)}$ are connected through two parts. Firstly, these two coefficients have the same sparsity, because the

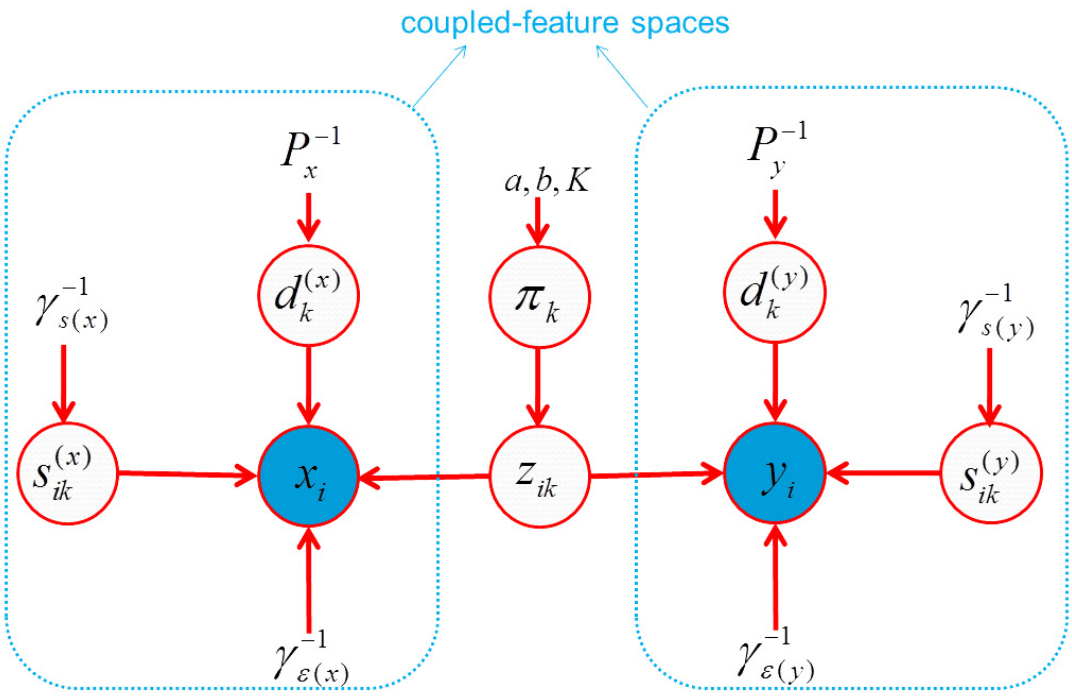



**Figure 4.1:** Graphical representation of the BP-JDL model for coupled feature spaces. $\mathbf{x}_i$ and $\mathbf{y}_i, i = 1, 2, \ldots, N$ are training samples for each feature space and we assume $\mathbf{x}_i = \mathbf{D}^{(x)}(\mathbf{z}_i \circ \mathbf{s}_i^{(x)}) + \epsilon_i^{(x)}$. For the coefficients $(\mathbf{z}_i \circ \mathbf{s}_i^{(x)})$, $\mathbf{z}_i$ is a binary vector $(z_{i1}, \ldots, z_{iK})$ that indicates which dictionary atoms are used by $\mathbf{x}_i$ and $\mathbf{s}_i^{(x)}$ is a vector $(s_{i1}^{(x)}, \ldots, s_{iK}^{(x)})$ of coefficient values. $\mathbf{d}_k^{(x)}$, $\mathbf{s}_i^{(x)}$ and $\epsilon_i^{(x)}$ are Gaussian distributed with variance $P_x^{-1}\mathbf{I}_{P_x}$, $\gamma_{s^{(x)}}^{-1}\mathbf{I}_K$ and $\gamma_{\epsilon^{(x)}}^{-1}\mathbf{I}_P$, respectively. Similar distribution is assumed for $\mathbf{d}_k^{(y)}$, $\mathbf{s}_i^{(y)}$ and $\epsilon_i^{(y)}$. $z_{ik}$ is Bernoulli distributed with parameter $\pi_k$ and $\pi_k$ is Beta distributed with parameters $\frac{a}{K}$ and $\frac{b(K-1)}{K}$.

same $z_i$ is used to construct both coefficients. Secondly, we use the mapping matrix $M$ to reveal the in-explicit relationship between coefficient values.

Finally, a graph representation of the BP-JDL is shown in Figure 4.1.

## 4.2 Gibbs-sampling Inference

Elements in Eq. 4.3 are in the conjugate exponential family, and therefore the posterior inference may be implemented via Gibbs-sampling method with analytic update equations. The joint distribution of BP-JDL is:

$$
\begin{aligned}
&P(\mathbf{X}, \mathbf{Y}, \mathbf{D}^{(x)}, \mathbf{D}^{(y)}, Z, S^{(x)}, S^{(y)}, \pi, \gamma_{s^{(x)}}, \gamma_{s^{(y)}}, \gamma_{\epsilon^{(x)}}, \gamma_{\epsilon^{(y)}}) \\
=&\prod_{i=1}^{N} \mathcal{N}(x_i; \mathbf{D}^{(x)}(z_i \circ s_i^{(x)}), \gamma_{\epsilon^{(x)}}^{-1} I_{P_x}) \mathcal{N}(y_i; \mathbf{D}^{(y)}(z_i \circ s_i^{(y)}), \gamma_{\epsilon^{(y)}}^{-1} I_{P_y}) \\
&\prod_{i=1}^{N} \mathcal{N}(s_i^{(x)}; 0, \gamma_{s^{(x)}}^{-1} I_K) \mathcal{N}(s_i^{(y)}, \gamma_{s^{(y)}}^{-1} I_K) \\
&\prod_{k=1}^{K} \mathcal{N}(\mathbf{d}_k^{(x)}; 0, P_x^{-1} I_{P_x}) \mathcal{N}(d_k^{(y)}; 0, P_y^{-1} I_{P_y}) Beta(\pi_k; a, b) \\
&\prod_{i=1}^{N} \prod_{k=1}^{K} Bernoulli(z_{ik}; \pi_k) \\
&\Gamma(\gamma_{s^{(x)}}; c, d) \Gamma(\gamma_{s^{(y)}}; c, d) \Gamma(\gamma_{\epsilon^{(x)}}; e, f) \Gamma(\gamma_{\epsilon^{(y)}}; e, f)
\end{aligned}
\tag{4.5}
$$

The Gibbs sampling update equations are:

- Sample $\mathbf{d}_k^{(x)}$

$$
p(\mathbf{d}_k^{(x)}|-) \sim \mathcal{N}(\mathbf{d}_k^{(x)}; 0, P_x^{-1} I_{P_x}) \prod_{i=1}^{N} \mathcal{N}(x_i; \mathbf{D}^{(x)}(z_i \circ s_i^{(x)}), \gamma_{\epsilon^{(x)}}^{-1} I_{P_x}) \tag{4.6}
$$

$\mathbf{d}_k$ can be drawn from a normal distribution

$$
p(\mathbf{d}_k^{(x)}|-) \sim \mathcal{N}(\mu_{\mathbf{d}_k^{(x)}}, \Sigma_{\mathbf{d}_k^{(x)}}) \tag{4.7}
$$

and

$$
\begin{aligned}
\Sigma_{\mathbf{d}_k^{(x)}} &= (P_x \mathbf{I} + \gamma_\epsilon^{(x)} \sum_{i=1}^{N} z_{ik}^2 {s_{ik}^{(x)}}^2)^{-1} \\
\mu_{\mathbf{d}_k^{(x)}} &= \gamma_\epsilon^{(x)} \Sigma_{\mathbf{d}_k^{(x)}} \sum_{i=1}^{N} z_{ik} s_{ik}^{(x)} \mathbf{x}_i^{-k}
\end{aligned}
\tag{4.8}
$$

where $\mathbf{x}_i^{-k} = \mathbf{x}_i - \mathbf{D}(\mathbf{s}_i^{(x)} \circ \mathbf{z}_i) + \mathbf{d}_k^{(x)}(s_{ik}^{(x)} \circ z_{ik})$.

- Sample $z_{ik}$

$$\begin{aligned} p(z_{ik}=1|-) \sim & \mathcal{N}(x_i; \mathbf{D}^{(x)}(z_i \circ s_i^{(x)}), \gamma_{\epsilon^{(x)}}^{-1} I_{P_x}) \\ & \mathcal{N}(y_i; \mathbf{D}^{(y)}(z_i \circ s_i^{(y)}), \gamma_{\epsilon^{(y)}}^{-1} I_{P_y}) \\ & Bernoulli(z_{ik}; \pi_k) \end{aligned} \tag{4.9}$$

The posterior probability of $z_{ik} = 1$ can be expressed as:

$$\begin{aligned} & p(z_{ik}=1|-) \\ \propto \quad & \pi_k \exp[ -\frac{\gamma_\epsilon^{(x)}}{2}({s_{ik}^{(x)}}^2 {\mathbf{d}_k^{(x)}}^T \mathbf{d}_k^{(x)} - 2 s_{ik}^{(x)} {\mathbf{d}_k^{(x)}}^T \mathbf{x}_i^{-k}) \\ & -\frac{\gamma_\epsilon^{(y)}}{2}({s_{ik}^{(y)}}^2 {\mathbf{d}_k^{(y)}}^T \mathbf{d}_k^{(y)} - 2 s_{ik}^{(y)} {\mathbf{d}_k^{(y)}}^T \mathbf{y}_i^{-k})] \end{aligned} \tag{4.10}$$

and the posterior probability of $z_{ik} = 0$ can be expressed as:

$$p(z_{ik}=0|-) = 1 - \pi_k \tag{4.11}$$

- Sample $s_{ik}^{(x)}$

$$p(\mathbf{s}_{ik}^{(x)}|-) \sim \mathcal{N}(x_i; \mathbf{D}^{(x)}(z_i \circ s_i^{(x)}), \gamma_{\epsilon^{(x)}}^{-1} I_{P_x}) \mathcal{N}(s_i^{(x)}; 0, \gamma_{s^{(x)}}^{-1} I_K) \tag{4.12}$$

$\mathbf{s}_{ik}$ can be drawn from a normal distribution

$$p(s_{ik}^{(x)}|-) \sim \mathcal{N}(\mu_{s_{ik}^{(x)}}, \Sigma_{s_{ik}^{(x)}}) \tag{4.13}$$

and

$$\begin{aligned} \Sigma_{s_{ik}^{(x)}} &= (\gamma_s^{(x)} + \gamma_\epsilon^{(x)} z_{ik}^2 {\mathbf{d}_k^{(x)}}^T \mathbf{d}_k^{(x)})^{-1} \\ \mu_{s_{ik}^{(x)}} &= \gamma_\epsilon^{(x)} \Sigma_{s_{ik}^{(x)}} (z_{ik} {\mathbf{d}_k^{(x)}}^T \mathbf{x}_i^{-k}) \end{aligned} \tag{4.14}$$

- Sample $\pi_k$

$$p(\pi_k|-) \ \sim Beta(\pi_k; a, b) \prod_{i=1}^{N} Bernoulli(z_{ik}; \pi_k) \tag{4.15}$$

$\pi_k$ can be drawn from a Beta distribution as

$$p(\pi_k|-) \sim Beta(\pi_k; a, b) \tag{4.16}$$

where

$$\begin{aligned} a =& \frac{a_0}{K} + \sum_{i=1}^{N} z_{ik} \\ b =& \frac{b_0(K-1)}{K} + N - \sum_{i=1}^{N} z_{ik} \end{aligned} \tag{4.17}$$

- Sample $\gamma_{s^{(x)}}$

$$p(\gamma_{s^{(x)}}|-) \sim \Gamma(\gamma_{s^{(x)}}; c, d) \prod_{i=1}^{N} \mathcal{N}(s_i^{(x)}; 0, \gamma_{s^{(x)}}^{-1} I_K) \tag{4.18}$$

$\gamma_{s^{(x)}}$ can be drawn from a Gamma distribution as

$$p(\gamma_{s^{(x)}}|-) \sim \Gamma(c + \frac{1}{2}KN, d + \frac{1}{2}\sum_{i=1}^{N} \|\mathbf{s}_i^{(x)T}\mathbf{s}_i^{(x)}\|) \tag{4.19}$$

- Sample $\gamma_{\epsilon}^{(x)}$

$$p(\gamma_{\epsilon^{(x)}}|-) \sim \Gamma(\gamma_{s^{(x)}}; e, f) \prod_{i=1}^{N} \mathcal{N}(x_i; \mathbf{D}^{(x)}(z_i \circ s_i^{(x)}), \gamma_{\epsilon^{(x)}}^{-1} I_{P_x}) \tag{4.20}$$

$\gamma_{\epsilon^{(x)}}$ can be drawn from a Gamma distribution as

$$p(\gamma_{\epsilon^{(x)}}|-) \sim \Gamma(e + \frac{1}{2}N, f + \frac{1}{2}\sum_{i=1}^{N} \|\mathbf{x}_i^{-k}\|^2) \tag{4.21}$$

The $\mathbf{d}_k^{(y)}$, $s_{ik}^{(y)}$, $\gamma_{s^{(y)}}$ and $\gamma_{\epsilon^{(y)}}$ can be sampled in similar way of $\mathbf{d}_k^{(x)}$, $s_{ik}^{(x)}$, $\gamma_{s^{(x)}}$ and $\gamma_{\epsilon^{(x)}}$, respectively.

# Chapter 5

# Application of Single Image Super-Resolution

## 5.1 Beta Process Dictionary Learning for Single Feature Space

We first evaluate the performance of the beta process dictionary learning for single feature space (BP) for single image super-resolution (SISR) from the quality of the dictionary generated as well as the fidelity of the high-resolution image. We compare the BP with two state-of-the-art single feature space dictionary learning algorithm: sparse coding based super-resolution (ScSR) and K-SVD.

In the experiment, the dictionaries are learned using the ScSR, the K-SVD and the BP, respectively. Dictionaries for factors of 2 and 3 magnification are learned and used for generating super-resolution images.

All dictionaries are trained from 100,000 patch pairs sampled from database provided in Yang et al. (2008). The patch pairs are only sampled from the luminance channel of the training images. For the pre-process, the low-resolution patches are upsampled to the same size as the high-resolution patches using bicubic interpolation. Because we only want to use the sparse representation to recover high frequency

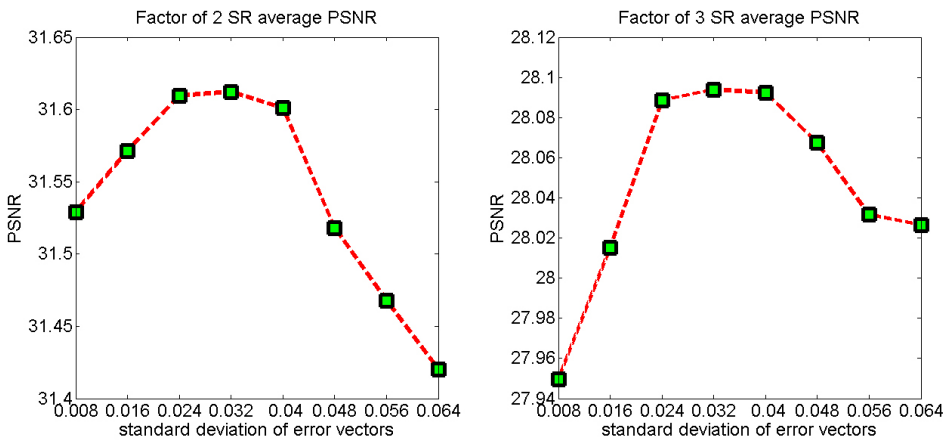



**Figure 5.1:** Super-resolution results of dictionaries learned using different standard deviation of error vectors. The max value of $\sigma = 0.064$ is the standard deviation of normalized training samples.

detail of the images, for the high-res patches, we subtract mean and normalize each patch. For the low-res patches, we extract the features from the low-res patches using Eq. 5.4 and normalize the features. Because in Yang et al. (2010) 1024 is found as the appropriate dictionary size to yield decent output, we set $K = 1024$ for all experiments of the ScSR and the K-SVD. We set the initial dictionary size $K$ of the BP as 1024, 2048 and 4096 to test the capability of the BP's $K$ inference. The ScSR, the K-SVD and the BP ran 40, 100 and 3000 iterations, respectively. For 3000 samples of the BP, the burn-in is 2500 samples and the dictionary is averaged using the rest 500 samples.

As we discussed in Chapter 3, we need to modify the beta process dictionary learning for the single feature space, by using a pre-defined $\sigma$ for noise variance instead of using a hyper-parameter. For the SR application, we choose $\sigma = 0.032$ based on SR test on 80 images as shown in Figure 5.1.

Although we use different methods to learn the dictionaries, we use the *same* method for super-resolution reconstruction to compare the effectiveness of the learned dictionaries. The SISR algorithm is summarized in Algorithm 1. First, 80 high-resolution images (8 categories, 10 images in each category) are blurred and down-sampled to $\frac{1}{2}$ and $\frac{1}{3}$ of the original size to produce the input low-resolution images.

Next, the high-resolution images are reconstructed using the Eq. 6.6 with fixed $\mathbf{D}$. In addition, images reconstructed using the Bicubic interpolation are compared as well. Figure 5.2 shows all test images.

**Algorithm 1** Single Image Super Resolution with BP dictionaries

**Input:** Low-res image $\mathbf{L}$, learned $\mathbf{D}^{(y)}$ and $\mathbf{D}^{(x)}$.
**Output:** High-res image $\mathbf{H}^*$
**Step 1** Sample low-res patch $l_i$ from the input image $\mathbf{L}$ with overlap $\omega$. Construct $\mathbf{y}_i$ using the four feature extraction operators. Learn $\alpha_i$ using the Efficient $\ell^1$:

$$\alpha_i = \arg\min_{\alpha_i} \frac{1}{2}\|\mathbf{D}^{(y)}\alpha_i - \mathbf{y}_i\|_2^2 + \lambda\|\alpha_i\|_1 \tag{5.1}$$

**Step 2** Recover the high-res patch $h_i$ using $\alpha$ and learned $\mathbf{D}^{(x)}$:

$$h_i = \mathbf{D}^{(x)}\alpha_i^{(x)} \tag{5.2}$$

After the recovery of all high-res patches, the initial high-res image $\mathbf{H}_0$ can be reconstructed with overlap $\omega$.
**Step 4** A global constraint is enforced to further improve the reconstruction accuracy:

$$\begin{aligned} &\mathbf{H}^* = \arg\min_{\mathbf{H}} \|\mathbf{H} - \mathbf{H}_0\|^2 \\ &\text{s.t.} \downarrow B\mathbf{H} = \mathbf{L} \end{aligned} \tag{5.3}$$

For the detail of the reconstruction algorithm, the $\lambda$ is set to 0.15 and the overlap is set to maximum value (patch size $-1$). We subtract mean and normalize each low-res patch as we did in the training. We also normalize the $\mathbf{D}_l$ before reconstruction.

### 5.1.1 Performance Metric

To evaluate dictionary learning results, we firstly show the dictionary size $K$ inferred by BP. Next, the learning time and the sparsity level are evaluated. The sparsity level is calculated by the average number of dictionary atoms used for all training samples. Moreover, we calculate the Root Mean Square (RMS) errors for the dictionaries and coefficients learned. Finally, the SR reconstruction results are evaluated via peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) Wang et al. (2004).

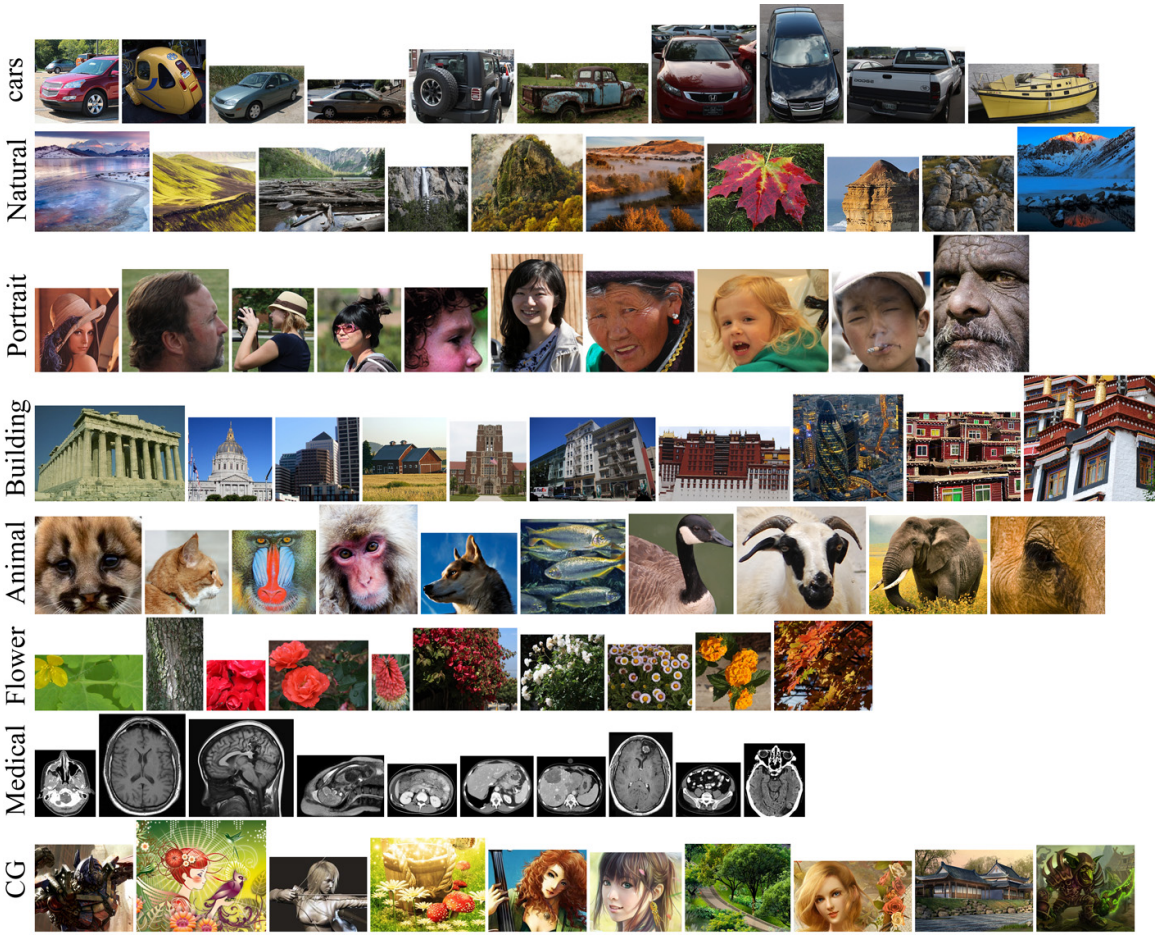



**Figure 5.2:** 80 test images for super-resolution. The images are divided into 8 categories, including car, natural, portrait, building, animal, flower, medical and CG. Each category has 10 test images.

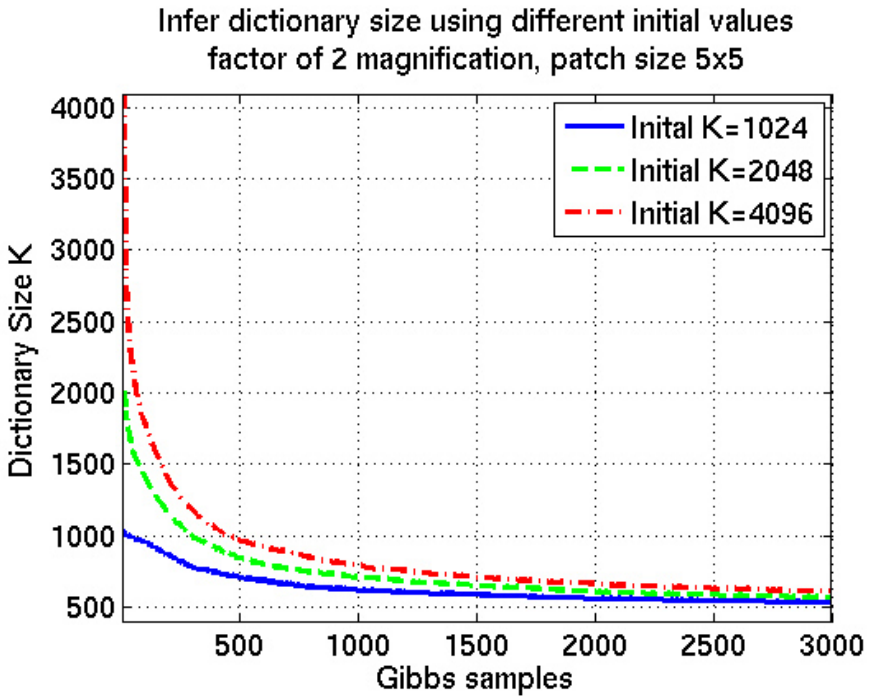



**Figure 5.3:** BP dictionary learning with different initial $K$.

Higher SSIM indicates more similar structure between the recovered image and the original image.

### 5.1.2 Dictionary Learning Results

Firstly, the dictionary size inferred by the BP is shown in Figure 5.12. With different initial $K$, the BP successfully inferred appropriate dictionary size. After the first 500 samples, the dictionary size are reduced to below 1000, and gradually converge to near 500 after 3000 samples.

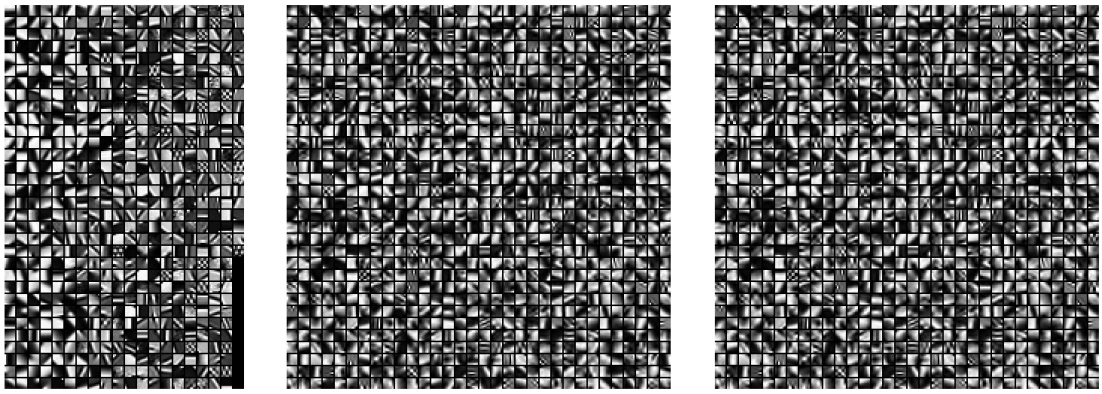
Next, tables 5.1 shows the dictionary learning results of factor of 2 and 3 magnification. With the initial dictionary size of 1024, the BP is able to infer 38.6% and 38.9% smaller size dictionaries for factor of 2 and 3 magnification, respectively. In addition, the BP method is able to infer appropriate sparsity level $T$ similar to the $\ell^1$ method without having to set the initial sparsity, confirming that the BP has the same sparseness property as the efficient $\ell^1$.

**Table 5.1:** Comparison of dictionary learning results. For the BP, the third column is the dictionary size $K$ inferred. The initial $K$ is set to 1024 for all experiments. The fifth column is the dictionary learning time (hours). The sixth column (Sparsity) is the average number of dictionary atoms used for $100,000$ training samples. For K-SVD, the initial $T_0$ in Eq. 2.7 is set to 20. Results were produced on a Dell T3500 Workstation with 2.66G Intel Xeon X5550 CPU and 12GB of RAM running Ubuntu and Matlab V7.12.0.

| Zoom | Method | Dictionary Size | Patch Size | Learning Time | Sparsity | Learning RMSE |
|---|---|---|---|---|---|---|
| 2× | BP | **629** | $7 \times 7$ | 15.4 | 10.4 | 0.28 |
| | $\ell^1$ | 1024 | $7 \times 7$ | 2.7 | 11.4 | 0.37 |
| | K-SVD | 1024 | $7 \times 7$ | 1.4 | 20.0 | 0.15 |
| 3× | BP | **626** | $7 \times 7$ | 15.4 | 10.4 | 0.28 |
| | $\ell^1$ | 1024 | $7 \times 7$ | 2.7 | 10.8 | 0.37 |
| | K-SVD | 1024 | $7 \times 7$ | 1.4 | 20.0 | 0.15 |

For the dictionary learning errors, the RMS errors of the BP are always smaller than those of the efficient $\ell^1$, indicating that the BP methods can reconstruct the training samples with less errors and less dictionary atoms. Although the K-SVD always has the least RMS errors after dictionary learning, the worse SR results of the K-SVD (Table 5.2) indicate that a dictionary with less dictionary learning RMS errors does not guarantee less super-resolution reconstruction errors. On the contrary, the K-SVD may suffer from the problem of overfitting the training samples. Finally, for the training time, although the BP has the slowest dictionary learning time, once dictionaries are learned, the BP dictionary will have a shorter SR reconstruction time because it has much less dictionary atoms. Table 5.3 shows the SR reconstruction time comparison of learned dictionaries. In addition, because of the slow convergence property of the Gibbs sampling, a variational Bayesian method might be used in the future for the BP method to shorten the inference time.

In addition, Figure 5.4 shows the factor of 2 magnification $\mathbf{D}_h$ learned by three methods.



(a) BP (629) (b) Efficient $\ell^1$ (1024) (c) K-SVD (1024)

**Figure 5.4:** Factor of 2 magnification dictionaries ($\mathbf{D}_h$) learned by the BP, the efficient $\ell^1$ and the K-SVD, respectively. The dictionary trained by the BP contains 629 atoms. Dictionaries trained by the efficient $\ell^1$ and the K-SVD contain 1024 atoms. Each atom is a $7 \times 7$ size image patch and is normalized for display purpose.

### 5.1.3 Single Image Super-Resolution Results

We show the super-resolution results from the perspective of SR recovery accuracy, image quality, reconstruction patch size, overlap and time.

#### Recovery Accuracy and Image Quality

Table 5.2 shows the super-resolution PSNR and SSIM (Wang et al., 2004) of factor of 2 and 3 SR. Firstly, we can see that three sparse representation based SR methods have a better SR reconstruction accuracy than bicubic interpolation. Secondly, even the training images are mostly flower images, the dictionaries learned by three methods are able to provide a better SR reconstruction accuracy for images in other categories as well. Finally, the BP dictionaries are able to produce on average better SR results (higher PSNR and SSIM) than the other two srSR methods in all categories while using smaller size dictionary.

Next, Figure 5.5 shows examples of factor of 2 and 3 SR result images for visual comparison. Same as the PSNR and SSIM results, the three srSR methods are able to produce better SR images than bicubic interpolation. From the zoom in view of the reconstructed high-res images, we can see that images reconstructed using BP

dictionaries have least artifacts compared to other methods, confirming that 629 and 626 size dictionaries learned by BP can generate same or better SR results than 1024-size dictionaries learned by the efficient $\ell^1$ and K-SVD.

Overall, the BP dictionaries produce the best SR image quality for both factor of 2 and 3 magnification tasks.

**Table 5.2:** Comparison of super-resolution results. For the BP, the second column is the dictionary size $K$ inferred. The initial $K$ is set to 1024 for all experiments. The super-resolution results of images in 8 categories are shown in the last 8 columns in averaged PSNR(dB) and SSIM. Same patch size ($7 \times 7$) is used for all three srSR methods.

| Zoom | Method | Measures | Car | Natural | Portrait | Building | Animal | Flower | Medical | CG |
|---|---|---|---|---|---|---|---|---|---|---|
| 2× | Bicubic | PSNR | 28.3850 | 28.1938 | 34.6556 | 27.1484 | 32.2022 | 31.0307 | 28.2993 | 28.3840 |
| | | SSIM | 0.8577 | 0.8073 | 0.9132 | 0.8529 | 0.8745 | 0.8851 | 0.8890 | 0.8459 |
| | | VIF | 4.9254 | 5.2561 | 5.7095 | 4.7674 | 5.5473 | 5.3593 | 5.2195 | 5.5511 |
| | | GSM | 0.9877 | 0.9867 | 0.9939 | 0.9852 | 0.9916 | 0.9908 | 0.9894 | 0.9880 |
| | BP | PSNR | **30.1537** | **29.4053** | **36.6430** | **28.7265** | **34.0089** | **32.9616** | **30.9459** | **30.2236** |
| | | SSIM | **0.8962** | **0.8560** | **0.9319** | **0.8935** | **0.9051** | **0.9172** | **0.9278** | **0.8922** |
| | | VIF | **6.3477** | **6.6527** | **6.8812** | **6.2129** | **6.9190** | **7.0848** | **6.8936** | **7.2089** |
| | | GSM | **0.9925** | **0.9917** | **0.9964** | **0.9904** | **0.9950** | **0.9948** | **0.9945** | **0.9930** |
| | $\ell^1$ | PSNR | 29.9676 | 29.3193 | 36.4177 | 28.5752 | 33.9101 | 32.7538 | 30.6890 | 30.0447 |
| | | SSIM | 0.8927 | 0.8543 | 0.9307 | 0.8909 | 0.9046 | 0.9149 | 0.9250 | 0.8879 |
| | | VIF | 6.2148 | 6.5988 | 6.8024 | 6.1118 | 6.8743 | 6.9796 | 6.7718 | 7.0916 |
| | | GSM | 0.9921 | 0.9916 | 0.9962 | 0.9901 | 0.9949 | 0.9946 | 0.9942 | 0.9928 |
| | K-SVD | PSNR | 29.9229 | 29.2908 | 36.3244 | 28.5071 | 33.8601 | 32.6733 | 30.5028 | 29.9427 |
| | | SSIM | 0.8916 | 0.8534 | 0.9300 | 0.8896 | 0.9044 | 0.9139 | 0.9234 | 0.8861 |
| | | VIF | 6.2003 | 6.5926 | 6.7828 | 6.0847 | 6.8594 | 6.9496 | 6.7203 | 7.0609 |
| | | GSM | 0.9920 | 0.9916 | 0.9962 | 0.9899 | 0.9949 | 0.9945 | 0.9940 | 0.9926 |
| 3× | Bicubic | PSNR | 25.7746 | 25.9456 | 31.7204 | 24.5664 | 29.2845 | 28.0966 | 25.0768 | 25.6630 |
| | | SSIM | 0.7460 | 0.6667 | 0.8495 | 0.7208 | 0.7754 | 0.7695 | 0.7816 | 0.7194 |
| | | VIF | 2.8821 | 2.9041 | 3.5272 | 2.6599 | 3.1932 | 3.0177 | 3.0877 | 3.1736 |
| | | GSM | 0.9775 | 0.9759 | 0.9884 | 0.9726 | 0.9838 | 0.9810 | 0.9786 | 0.9767 |
| | BP | PSNR | **26.7424** | **26.5670** | **33.1723** | **25.4514** | **30.4074** | **29.2101** | **26.6701** | **26.6144** |
| | | SSIM | **0.7868** | **0.7109** | **0.8729** | **0.7653** | **0.8082** | **0.8101** | **0.8305** | **0.7666** |
| | | VIF | **3.5736** | **3.4974** | **4.1522** | **3.3005** | **3.7961** | **3.8068** | **3.8897** | **3.9208** |
| | | GSM | **0.9825** | **0.9815** | **0.9914** | **0.9787** | **0.9879** | **0.9862** | **0.9848** | **0.9824** |
| | $\ell^1$ | PSNR | 26.6898 | 26.5423 | 33.0765 | 25.3900 | 30.3414 | 29.1372 | 26.5976 | 26.5605 |
| | | SSIM | 0.7865 | 0.7107 | 0.8718 | 0.7647 | 0.8066 | 0.8089 | 0.8305 | 0.7651 |
| | | VIF | 3.5305 | 3.4678 | 4.1367 | 3.2429 | 3.7798 | 3.7673 | 3.8731 | 3.8824 |
| | | GSM | 0.9823 | 0.9811 | 0.9913 | 0.9783 | 0.9877 | 0.9859 | 0.9846 | 0.9822 |
| | K-SVD | PSNR | 26.5507 | 26.5105 | 32.9172 | 25.3253 | 30.2439 | 29.0680 | 26.4083 | 26.4672 |
| | | SSIM | 0.7799 | 0.7124 | 0.8698 | 0.7628 | 0.8075 | 0.8080 | 0.8235 | 0.7628 |
| | | VIF | 3.4430 | 3.4339 | 4.0822 | 3.1911 | 3.7421 | 3.6997 | 3.7669 | 3.8094 |
| | | GSM | 0.9815 | 0.9809 | 0.9910 | 0.9778 | 0.9875 | 0.9855 | 0.9838 | 0.9816 |

### Patch Size and Overlap

Next, we study the relationship between the high-resolution patch size and SR reconstruction accuracy and quality. We test the SR PSNR and SSIM using patch size $5 \times 5$, $7 \times 7$ and $9 \times 9$ for three srSR methods. The average PSNR and SSIM of facotr of 3 SR images in 8 different categories are shown in Figures 5.6 and 5.7. For the efficient $\ell^1$ and BP, both SR PSNR and SSIM of patch size $7 \times 7$ are mostly better than patch size $5 \times 5$ and $9 \times 9$. For the K-SVD, 75% of reconstructed images have a better PSNR and SSIM using patch size $9 \times 9$. In all, BP provide the best SR PSNR and SSIM using different patch size compared to the other two methods. In addition, on average SR using $7 \times 7$ dictionaries generate the best SR results.

For the overlap mentioned in Algorithm 1, theatrically the more overlap we use during the SR reconstruction process, the better the SR results are. From the test result, we confirm that if the larger the overlap value, the better the PSNR and SSIM of reconstructed image. An example of the relationship between the overlap value and PSNR and SSIM is shown in Figure 5.8. From the results we can see that we got the best PSNR and SSIM for all three srSR methods when the maximum overlap is used (patch size - 1). Overall, BP dictionaries still generate better results than the efficient $\ell^1$ and K-SVD.

### Super-Resolution Time

Finally, the average super-resolution time is shown in Table 5.3. From the results we can see that SR using BP dictionaries indeed benefit from a smaller dictionary size. The SR time of the BP dictionaries is on average 34% and 26% shorter than the SR time of the efficient $\ell^1$ dictionaries and the K-SVD dictionaries, respectively. This advantage of BP is critical when one want to use the srSR on energy constraint applications such as mobile application or wireless sensor network application.

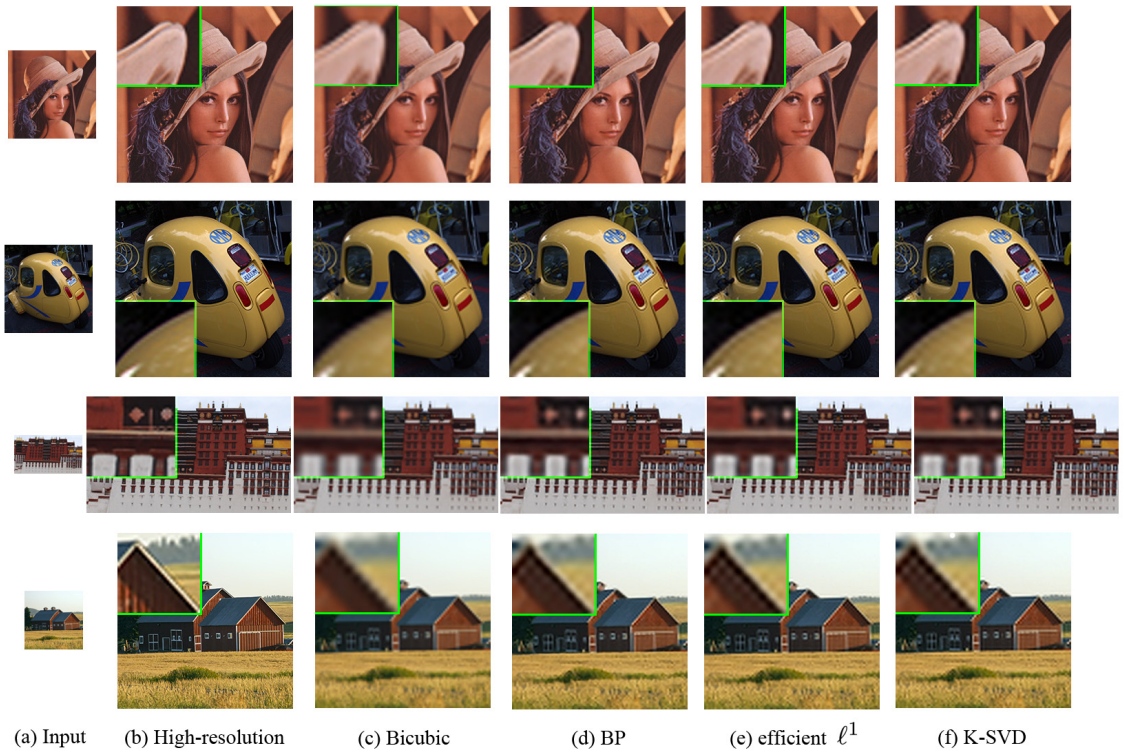



**Figure 5.5:** Comparison of super-resolution images reconstructed using the Bicubic, the BP, the efficient $\ell^1$ and the K-SVD, respectively. (a) low-resolution input images. (b) original high-resolution images used to create the low-resolution images. The upper two rows show the factor of 2 magnification results. The lower two rows show the factor of 3 magnification results. Generally, the sparse representation based SR is better than the Bicubic interpolation. The BP dictionary produces the best SR image quality.

**Table 5.3:** Comparison of average super-resolution reconstruction time (seconds). $K$ is dictionary size. The patch size is $7 \times 7$. The SR time of the BP dictionaries is on average 34% and 26% shorter than the SR time of the efficient $\ell^1$ dictionaries and the K-SVD dictionaries, respectively.

| Zoom | Method | K | Car | Natural | Portrait | Building | Animal | Flower | Medical | CG |
|---|---|---|---|---|---|---|---|---|---|---|
| 2× | BP | **629** | **132** | **188** | **188** | **183** | **216** | **126** | **98** | **191** |
| | $\ell^1$ | 1024 | 191 | 256 | 280 | 308 | 328 | 180 | 157 | 279 |
| | K-SVD | 1024 | 187 | 251 | 253 | 256 | 270 | 154 | 122 | 233 |
| 3× | BP | **626** | **145** | **203** | **202** | **199** | **211** | **124** | **97** | **185** |
| | $\ell^1$ | 1024 | 189 | 298 | 302 | 333 | 302 | 198 | 163 | 301 |
| | K-SVD | 1024 | 215 | 283 | 288 | 243 | 298 | 170 | 147 | 268 |

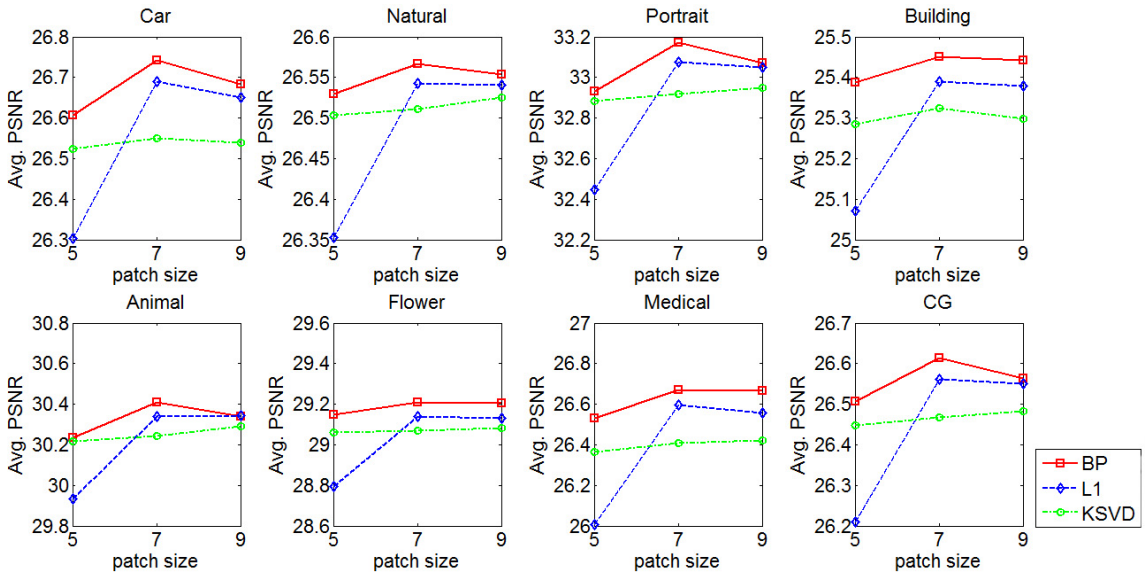



**Figure 5.6:** Super-resolution reconstruction PSNR of different size of high-res patch. The average PSNR of 8 categories are shown in individual sub-figures.

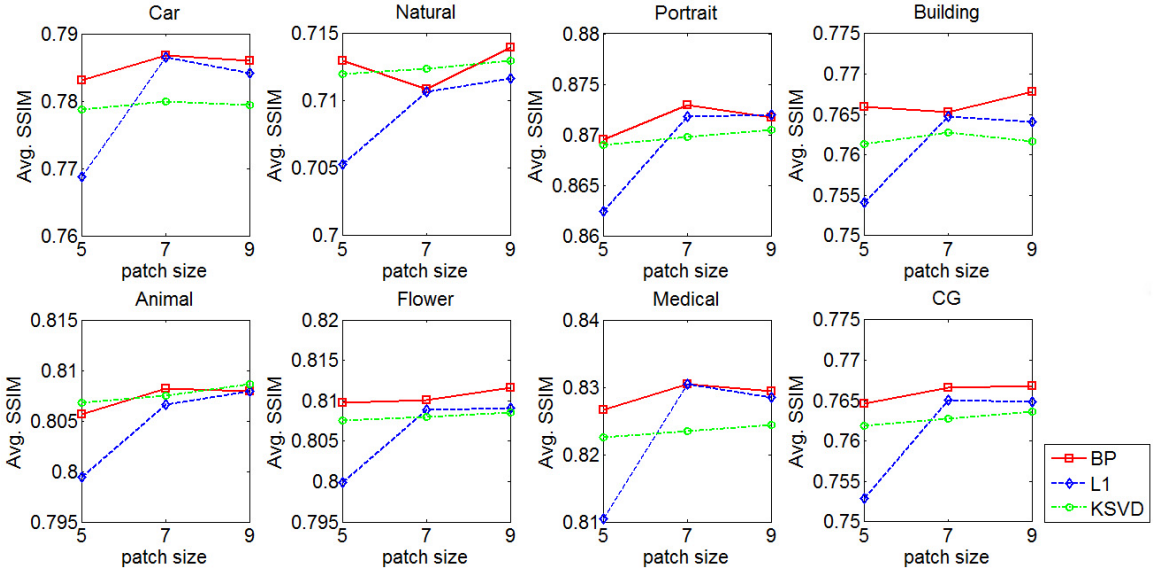



**Figure 5.7:** Super-resolution reconstruction SSIM of different size of high-res patch. The average SSIM of 8 categories are shown in individual sub-figures.

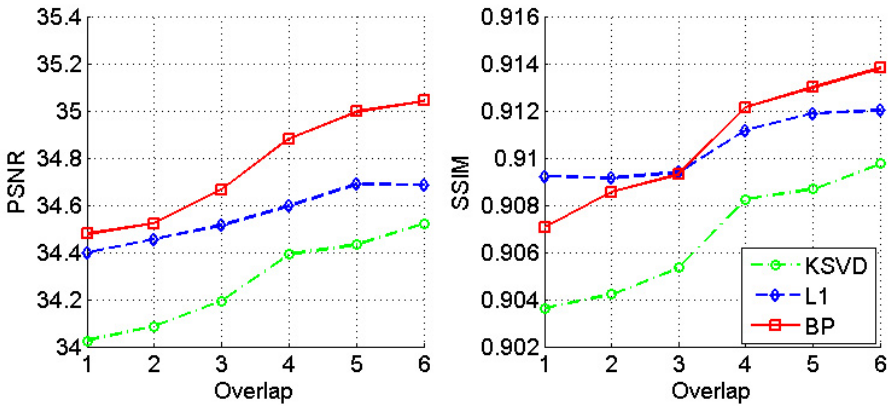



**Figure 5.8:** Test results of different overlap during super-resolution of the *Lena* image. The patch size is $7 \times 7$.

## 5.2 Beta Process Joint Dictionary Learning

Next, we evaluate the proposed BP-JDL for coupled feature spaces algorithm for the SISR. The two feature spaces are constructed as:

$$\begin{aligned} \mathbf{x}_i &= h; \\ \mathbf{y}_i &= [F_1 l; F_2 l; F_3 l; F_4 l] \end{aligned} \tag{5.4}$$

We use the proposed BP-JDL method to learn $\mathbf{D}^{(x)}$, $\mathbf{D}^{(y)}$ and the mapping matrix $\mathbf{M}$ for the two feature spaces. Similar to Chapter 3, the variance of error vectors of BP-JDL are set to constant instead of Inverse-gamma distributed, because the the distribution of error vectors, although close to is not exactly Gaussian. If we still use the inverse-Gamma distribution for the variance of error vectors, the Gaussian and inverse-Gamma model cannot fit the data well during the learning process. Therefore, a constant variance of error vectors is used to provide a lower bound for the variance of the error vectors. In this way, we can learn the dictionary successfully. In order to find optimal value of variance of error vectors, we tested different variance values that equal to $1\% \sim 50\%$ of data variance. In addition, because in the BP-JDL, we need to specify error vector variances for both feature spaces, we set them to equal value for convenience. Based on the super-resolution results of training images, we found

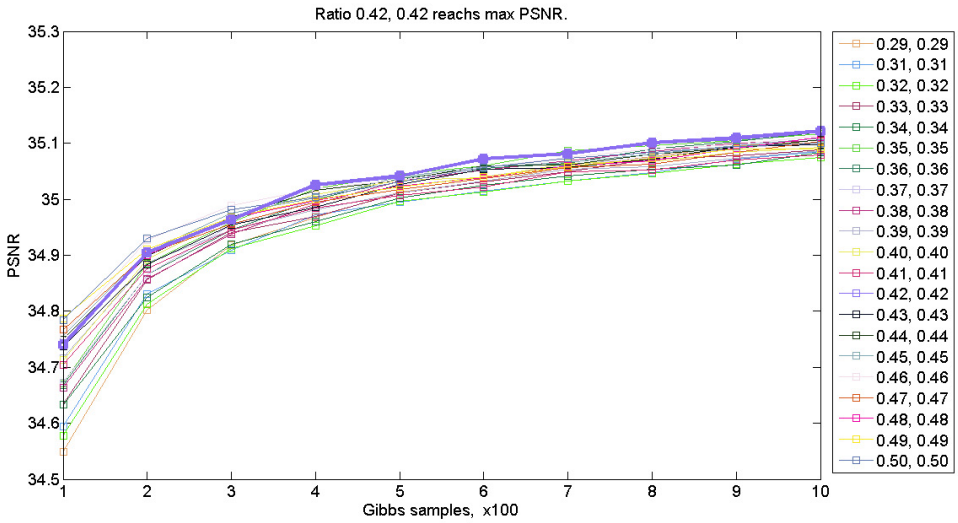



**Figure 5.9:** Super-resolution results of dictionaries learned using different noise variance ratios. We set the same noise variance ratio for both feature spaces.

that set both error variance equal to 0.42∗(data variance)the algorithm generate the best PSNR. Part of test results is shown in Figure 5.9. The BP-JDL runs 1000 Gibbs samples on different noise variance ratios.

Once the dictionaries are learned, we can use them for super-resolution reconstruction. The single image super-resolution reconstruction can be carried out in four steps. The first step calculates the sparse coding of observed low-res feature using learned low-res feature dictionary. In order to compare our dictionary with dictionaries learned by (Yang et al., 2010, 2012a; Wang et al., 2012), we use the standard $\ell^1$ sparse coding method for step 1 (Lee et al., 2007). The second step maps the sparse coding of the low-res feature to sparse coding of the high-res feature using the learned matrix $\mathbf{M}$. The third step recovers the high-res patch using the learned high-res feature dictionary. Because we do not directly use the low-res patch in Eq. 5.4, the reconstructed high-res image $\mathbf{H}_0$ may not satisfy the constraint $\downarrow B\mathbf{H} = \mathbf{L}$, thus the last step enforces a global constraint to eliminate this inconsistency by projecting $\mathbf{H}_0$ onto the solution space of $\downarrow B\mathbf{H} = \mathbf{L}$. In addition, because the recently introduced non-local redundancies in image are useful for image restoration (Buades et al., 2005;

Dabov et al., 2007a), we also incorporate the non-local self-similarities in step 4. The four steps are summarized in Algorithm 2.

**Algorithm 2** Single Image Super Resolution with BP-JDL dictionaries

**Input:** Low-res image $\mathbf{L}$, learned $\mathbf{D}^{(x)}$, $\mathbf{D}^{(y)}$ and $\mathbf{M}$
**Output:** High-res image $\mathbf{H}^*$
**Step 1** Sample low-res patch $l_i$ from the input image $\mathbf{L}$ with overlap $\omega$. Construct $\mathbf{y}_i$ using the four feature extraction operators. Learn $\alpha_i^{(y)}$ using the $\ell^1$ sparse coding:

$$\alpha_i^{(y)} = \arg\min_{\alpha_i^{(y)}} \frac{1}{2}\|\mathbf{D}^{(y)}\alpha_i^{(y)} - \mathbf{y}_i\|_2^2 + \lambda\|\alpha_i^{(y)}\|_1 \tag{5.5}$$

**Step 2** Map the sparse coefficients $\alpha^{(y)}$ to $\alpha^{(x)}$ using the learned $\mathbf{M}$:

$$\alpha_i^{(x)} = \mathbf{M}\mathbf{z}_i\alpha_i^{(y)} \tag{5.6}$$

where $\mathbf{z}_i$ is a binary vector that $z_{ik} = 1$ if $\alpha_{ik}^{(y)} \neq 0$.
**Step 3** Recover the high-res patch $h_i$ using $\alpha^{(x)}$ and learned $\mathbf{D}^{(x)}$:

$$h_i = \mathbf{D}^{(x)}\alpha_i^{(x)} \tag{5.7}$$

After the recovery of all high-res patches, the initial high-res image $\mathbf{H}_0$ can be reconstructed with overlap $\omega$.
**Step 4** A global constraint and a non-local similarity constrain are enforced to further improve the reconstruction accuracy:

$$\begin{aligned} &\mathbf{H}^* = \arg\min_{\mathbf{H}} \|\mathbf{H} - \mathbf{H}_0\|^2 \\ &\text{s.t.} \downarrow B\mathbf{H} = \mathbf{L}, \|h_i - \sum_{m=1}^{M} b^m h_{i(0)}^m\|_2^2 \leq \epsilon \end{aligned} \tag{5.8}$$

where $h_i$ and $h_{i(0)}$ are patches in $\mathbf{H}$ and $\mathbf{H_0}$, respectively. $h_{i(0)}^m$ is the $m^{th}$ most similar patch to $h_{i(0)}$ and $b^m$ is the non-local weight defined in (Buades et al., 2005).

Eq. 5.8 can be solved by back projection method introduced in (Capel, 2001).

### 5.2.1 Experimental Design

We evaluate the performance of the proposed BP-JDL method when applied to single image super-resolution from perspectives of both the quality and the fidelity of

the high-resolution image. We compared our results with state-of-the-art dictionary learning based SISR method, including ScSR (Yang et al., 2008), Zeyde (Zeyde et al., 2010), SCDL (Wang et al., 2012), Bilevel Yang et al. (2012a), and the BP method we proposed in Chapter 3.

Dictionaries for factors of 2 and 3 magnification are learned and used for generating super-resolution images. The low-resolution patches are upsampled to the same size as the high-resolution patches. All dictionaries are trained from 100,000 patch pairs sampled from 10 categories of representative and texture rich images, as shown in Figure 5.11. The patch pairs are only sampled from the luminance channel of the training images because human eyes are more sensitive to luminance changes. The pre-process step is as same as we described in 5.1.

Before we proceed to the dictionary learning step, we pre-process the patch pairs by deleting the non-informative noise patches. According to (Olshausen and Fieldt, 1996), natural images contain localized, oriented, and bandpass structures, which cannot be characterized in term of linear, pairwise correlations. Localized structures, such as a step edge, its phases aligned across different spatial frequencies. Oriented structures, such as lines and edges, will also evade pairwise correlations because they require at least three-point statistics to characterize. Finally, bandpass structures in natural images will tend to produce local phase alignments in spatial frequency. Because we only want to learn patches with these characteristics, can we delete "other" patches even before the learning process?

Inspired by (Yang et al., 2012b), we can remove these non-informative patches using a threshold on the dominate measure (Edelman, 1988). For each high-res patch $h$, we firstly calculate its gradient map $G = [g_1, \cdot, g_n]^T$, where $g_i = [(\partial h(x,y)/\partial x), (\partial h(x,y)/\partial y)]$ is the $i$th pixels in the patch and $n$ is the total number of pixels in the patch. Next, we perform an SVD on $G$ to obtain $G = USV_T$. If the dominate measure of the patch

$$R = \frac{S_{1,1} - S_{2,2}}{S_{1,1} + S_{2,2}} \tag{5.9}$$

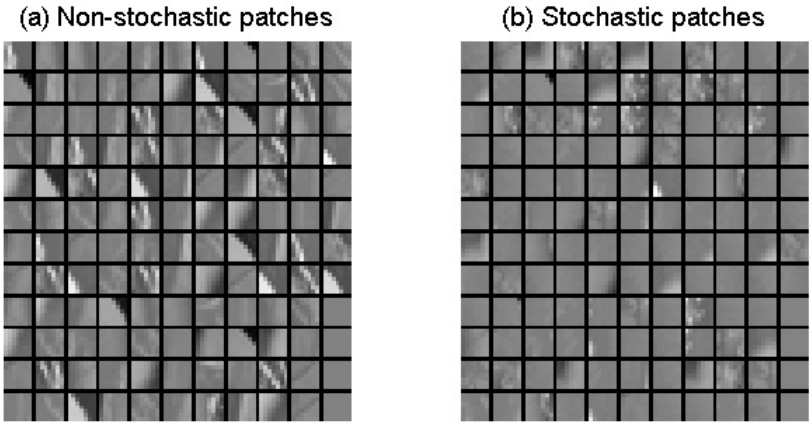



**Figure 5.10:** (a) Non-stochastic patches and (b) Stochastic patches. Patch size is $7 \times 7$.

is smaller than a threshold $R^*$, the patch is considered as stochastic (non-informative) patch. We consider these patches are not localized, oriented and bandpass patches and remove corresponding patch pairs for pre-processing. An example of deleted patches and remain patches for the training is shown in Figure 5.10. In the super-resolution application, we set $R^* = 0.27$ and all patches with dominate measure smaller than $R^*$ is removed. By removing these stochastic patches, the learned dictionary do not have to recover these stochastic patches, therefore the quality of the dictionary is improved.

For the dictionary learning, we set the initial dictionary size $K$ of BP-JDL as 1024, 2048 and 4096 to test the capability of BP-JDL's $K$ inference. We use 10000 Gibbs samples for BP-JDL, where the burn-in is 9500 samples and the dictionary is averaged using the rest 500 samples.

For the super-resolution reconstruction, high-resolution test images are blurred and down-sampled to 1/4 and 1/9 of the original size to produce the input low-resolution images. The high-resolution images are reconstructed using Algorithm 1


**Figure 5.11:** 10 training images.

with $\lambda$ set to 0.15 and the overlap set to its maximum value (i.e., patch size $-$ 1). In addition, images reconstructed using the Bicubic interpolation are compared as well.

### 5.2.2 Dictionary Learning Results

Firstly, dictionaries in coupled feature spaces are learned using the proposed BP-JDL algorithm. Compared to the dictionaries learned in concatenated spaces (Yang et al., 2010), the dictionaries learned by BP-JDL are able to reduce the learning root-mean-square (RMS) errors of high-res feature space $\mathcal{X}$ and low-res feature space $\mathcal{Y}$ by 27.5% and 40.5%, respectively. This result confirms that BP-JDL is capable to learn dictionaries that fit the data better by allowing the different coefficients values for the two spaces.

Secondly, the dictionary size inferred by BP-JDL is shown in Figure 5.12. Because BP-JDL has the non-parametric advantage, with different initial $K$s, the dictionary size decreases rapidly during the first 1000 samples and gradually converges to similar values, confirming that BP-JDL can infer appropriate dictionary size no matter what the initial value is. With the initial size of 1024, the BP-JDL inferred that $K = 771$ is an appropriate dictionary size. If we fix the dictionary size to 1024 for BP-JDL, the learning RMS errors and sparsity level of the 1024-size dictionaries stay the same as the 771-size dictionaries, indicating that 771 is the appropriate dictionary size for the training data. If the dictionary size is unknown, normally we need exhaustively

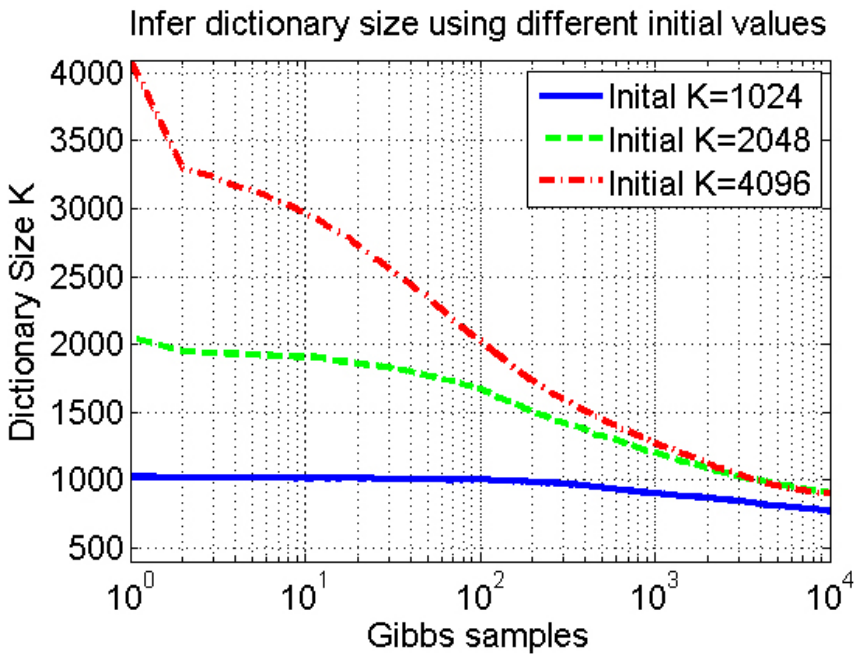



**Figure 5.12:** BP-JDL infers dictionary size non-parametrically.

search for the optimal size. Yang (Yang et al., 2010) found that the the 1024-size dictionary is optimal, however, the 771-size dictionary may have the same super-resolution performance as the 1024-size dictionary. Besides, since super-resolution using a smaller size dictionary needs less computational power, it may significantly affect the speed and energy consumption of super-resolution applications in resource-constrained environments.

Finally, a learned mapping matrix for 771-size dictionary is shown in Figure 5.13. This mapping matrix represent the implicit relationship between two feature spaces. We notice that the diagonal component is obvious and conclude that it's majorly an one-on-one mapping.

### 5.2.3 Single Image Super-Resolution Results

We evaluate the super-resolution (SR) results thoroughly via four image quality metrics, including peak signal-to-noise ratio (PSNR), structural similarity (SSIM) Wang et al. (2004), visual information fidelity (VIF) (Sheikh and Bovik, 2006) and gradient

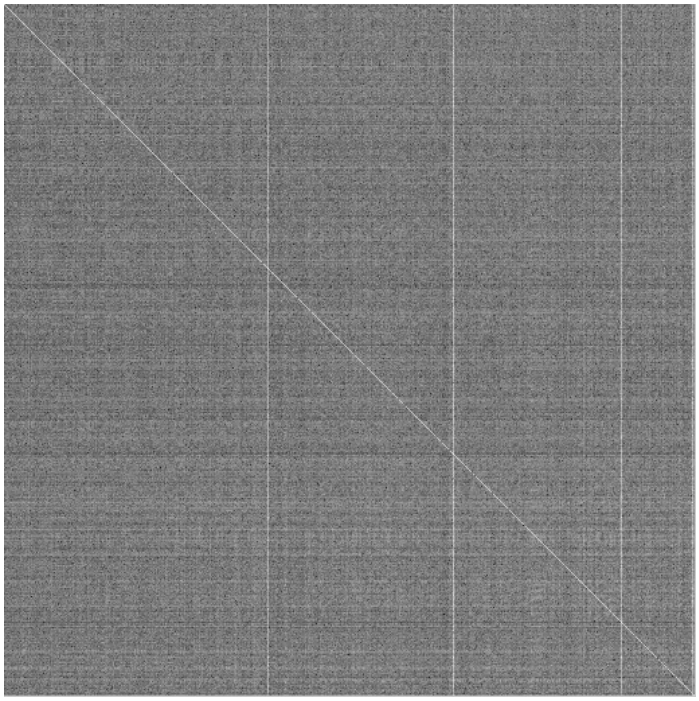


**Figure 5.13:** BP-JDL learned mapping matrix log(**M**) for 771-size dictionary.

**Table 5.4:** Comparison of factor of 2 magnification super-resolution results.

| Image | Measures | Bicubic | ScSR | BP | Zeyde | SCDL | Bilevel | BP-JDL |
|---|---|---|---|---|---|---|---|---|
| Lena | PSNR(dB) | 32.7947 | 34.6874 | 35.0411 | 34.2640 | 35.1311 | 35.0680 | **35.3308** |
| | SSIM | 0.8872 | 0.9120 | 0.9138 | 0.9044 | 0.9140 | 0.9130 | **0.9160** |
| | VIF | 5.3033 | 6.5612 | 6.6950 | 6.6390 | 6.6975 | 6.6653 | **6.7314** |
| | GSM | 0.9940 | 0.9964 | 0.9967 | 0.9961 | 0.9967 | 0.9967 | **0.9968** |
| Mountain | PSNR(dB) | 29.6999 | 31.2343 | 31.4006 | 31.0867 | 31.4010 | 31.3757 | **31.5459** |
| | SSIM | 0.8430 | 0.8909 | 0.8937 | 0.8874 | 0.8937 | 0.8918 | **0.8969** |
| | VIF | 5.6668 | 7.2328 | 7.2973 | 7.1961 | 7.3018 | 7.3059 | **7.3428** |
| | GSM | 0.9882 | 0.9929 | 0.9932 | 0.9926 | 0.9933 | 0.9932 | **0.9934** |
| House | PSNR(dB) | 26.3549 | 27.4334 | 27.5953 | 27.3055 | 27.6055 | 27.6115 | **27.7919** |
| | SSIM | 0.8048 | 0.8456 | 0.8495 | 0.8450 | 0.8510 | 0.8482 | **0.8525** |
| | VIF | 3.7844 | 4.5802 | 4.6516 | 4.5358 | 4.6646 | 4.6883 | **4.6963** |
| | GSM | 0.9834 | 0.9884 | 0.9888 | 0.9881 | 0.9889 | 0.9889 | **0.9890** |
| Lion | PSNR(dB) | 30.9312 | 32.5090 | 32.6021 | 32.4216 | 32.6028 | 32.5993 | **32.8818** |
| | SSIM | 0.8439 | 0.8941 | 0.8944 | 0.8926 | 0.8940 | 0.8929 | **0.8979** |
| | VIF | 5.6896 | 7.0111 | 7.0302 | 6.9814 | 7.0543 | 7.0510 | **7.0928** |
| | GSM | 0.9898 | 0.9941 | 0.9941 | 0.9939 | 0.9942 | 0.9941 | **0.9943** |
| Car | PSNR(dB) | 30.5383 | 32.5275 | 32.6720 | 32.3576 | 32.5904 | 32.8914 | **33.1157** |
| | SSIM | 0.9138 | 0.9381 | 0.9396 | 0.9370 | 0.9396 | 0.9419 | **0.9436** |
| | VIF | 4.7268 | 5.9447 | 6.0732 | 5.7780 | 6.0604 | 6.1172 | **6.2286** |
| | GSM | 0.9905 | 0.9939 | 0.9943 | 0.9939 | 0.9945 | 0.9946 | **0.9948** |

**Table 5.5:** Comparison of factor of 3 magnification super-resolution results.

| Image | Measures | Bicubic | ScSR | BP | Zeyde | SCDL | Bilevel | BP-JDL |
|---|---|---|---|---|---|---|---|---|
| Lena | PSNR(dB) | 30.0986 | 31.5125 | 31.5313 | 30.9077 | 31.5900 | 31.5808 | **31.6818** |
| | SSIM | 0.8019 | 0.8354 | 0.8355 | 0.8156 | 0.8347 | 0.8344 | **0.8377** |
| | VIF | 3.1540 | 3.8340 | 3.8910 | 3.7150 | 3.9560 | 3.4550 | **4.0070** |
| | GSM | 0.9885 | 0.9914 | 0.9916 | 0.9861 | 0.9919 | 0.9909 | **0.9920** |
| Mountain | PSNR(dB) | 27.0522 | 28.0436 | 28.0090 | 27.8258 | 28.0490 | 28.0606 | **28.1259** |
| | SSIM | 0.7000 | 0.7596 | 0.7581 | 0.7520 | 0.7607 | 0.7561 | **0.7636** |
| | VIF | 3.1975 | 3.9081 | 3.9294 | 3.7909 | 3.9492 | 3.3987 | **4.0154** |
| | GSM | 0.9778 | 0.9831 | 0.9833 | 0.98 | 0.9836 | 0.9813 | **0.9838** |
| House | PSNR(dB) | 24.4172 | 25.0136 | 25.0301 | 24.7198 | 25.0100 | 25.0277 | **25.0592** |
| | SSIM | 0.6881 | 0.7230 | 0.7235 | 0.7234 | 0.7236 | 0.7235 | **0.7248** |
| | VIF | 2.1192 | 2.5725 | **2.6088** | 2.5023 | 2.5938 | 2.3119 | 2.5968 |
| | GSM | 0.9729 | 0.9781 | 0.9787 | 0.9739 | **0.9785** | 0.9759 | 0.9783 |
| Lion | PSNR(dB) | 28.3921 | 29.0637 | 29.0025 | 29.0455 | 29.0483 | 29.1161 | **29.2190** |
| | SSIM | 0.7058 | 0.7496 | 0.7478 | 0.7498 | 0.7512 | 0.7473 | **0.7537** |
| | VIF | 3.1643 | 3.6576 | 3.6698 | 3.6063 | 3.7088 | 3.2403 | **3.7550** |
| | GSM | 0.9811 | 0.985 | 0.9852 | 0.9833 | 0.9856 | 0.9835 | **0.9857** |
| Car | PSNR(dB) | 27.4234 | 28.6083 | 28.6374 | 28.5011 | 28.4892 | 28.7231 | **28.8557** |
| | SSIM | 0.8259 | 0.8630 | 0.8645 | 0.8573 | 0.8635 | 0.8652 | **0.8673** |
| | VIF | 2.8075 | 3.4413 | 3.5084 | 3.3443 | 3.4771 | 3.1024 | **3.5621** |
| | GSM | 0.9816 | 0.9854 | 0.9857 | 0.9832 | 0.9857 | 0.9854 | **0.9862** |

similarity (GSM) (Liu et al., 2012). Higher SSIM values indicate more similar structure between the recovered image and the original image. Higher VIF values indicate a better visual quality of recovered image which is strong related to the relative image information. Higer GSM values indicate more similar gradient between recovered image and the original image.

Factors of 2 and 3 SR results are shown in Tables 5.4 and 5.5, respectively. In addition, the visual comparison example of factors of 2 and 3 SR results are shown in Fig. 5.15 and Fig. 5.16, respectively.

From the for image quality metrics comparison results (PSNR, SSIM, VIF and GSM), firstly we notice that sparse representation based SR methods generally perform better than the interpolation based method (e.g., bicubic), because the over-complete dictionaries can recover high-frequency details of images more accurately. Next, Zeyde's (Zeyde et al., 2010) two-step learned dictionaries have the similar performance as the coupled learned dictionaries (ScSR) (Yang et al., 2010), while

the BP and the most recent semi-coupled dictionary learning methods SCDL (Wang et al., 2012) and Bilevel (Yang et al., 2012a) outperform the coupled dictionary learning algorithm. Finally, the proposed BP-JDL method further pushes the limit by providing a flexible and consistent learning model, and is able to provide high-res images with the best recover accuracy (PSNR), structure similarity (SSIM), information fidelity (VIF) and gradient similarity (GSM).

From the visual comparison results, we also notice that generally sparse representation based SR methods produce sharper image than bicubic interpolation. Next, we notice the improvement of BP, SCDL and Bilevel methods compared to the ScSR method in terms of artifacts on the edges. Among the results of all sparse representation based methods, images produced by the proposed BP-JDL algorithm have the least artifacts, indicating that the proposed method can better restore the high-res images from low-res images.

During the SR reconstruction process, theoretically the more overlap of patches, the better the SR results. SR results of different overlap values are shown in Fig. 5.14. The results demonstrate the positive relationship between the overlap size and PSNR (SSIM), confirming using maximum overlap (patchsize - 1) can generate the best reconstruction results.

The average factor of 2 SR reconstruction time of ScSR, BP, Zeyde, SCDL, Bilevel and BP-JDL are 217.9s, 214.0s, 1.9s, 1837.8s, 218.7s and 213.5s, respectively. Results were produced on a Dell T3500 Workstation with 2.66G CPU and 12GB RAM running Matlab V7.12.0. Among these methods, the Zeyde method is the fastest. Although BP-JDL benefits from using a smaller dictionary compared to ScSR, the extra operation of Eq. 6.7 consumes extra time. However, BP-JDL is still faster than ScSR, SCDL and Bilevel methods. SCDL is the slowest method because it needs 32 dictionaries (clusters) for each feature space instead of single dictionary, thus consuming much more time than other methods.

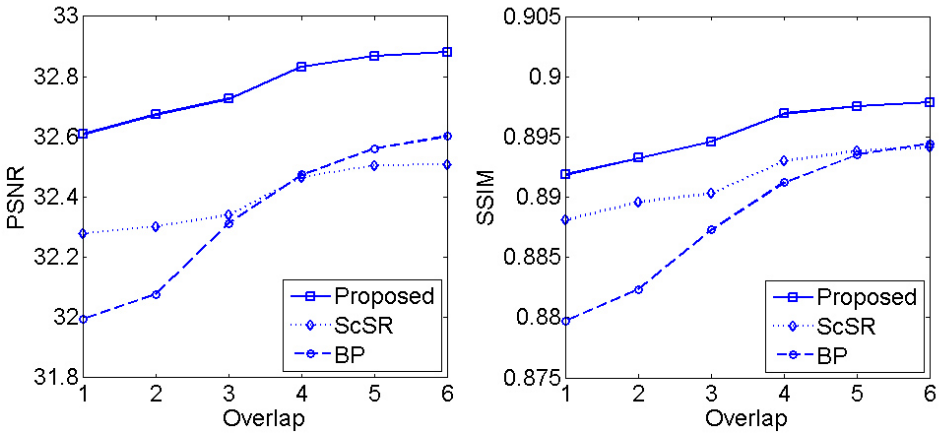



**Figure 5.14:** Effect of the overlap parameter on PSNR and SSIM of test image Lion.


**Figure 5.15:** Visual comparison of factor of 2 super-resolution results. The upper row shows the SR results of the image Lena. The lower row shows the SR results of the image House.

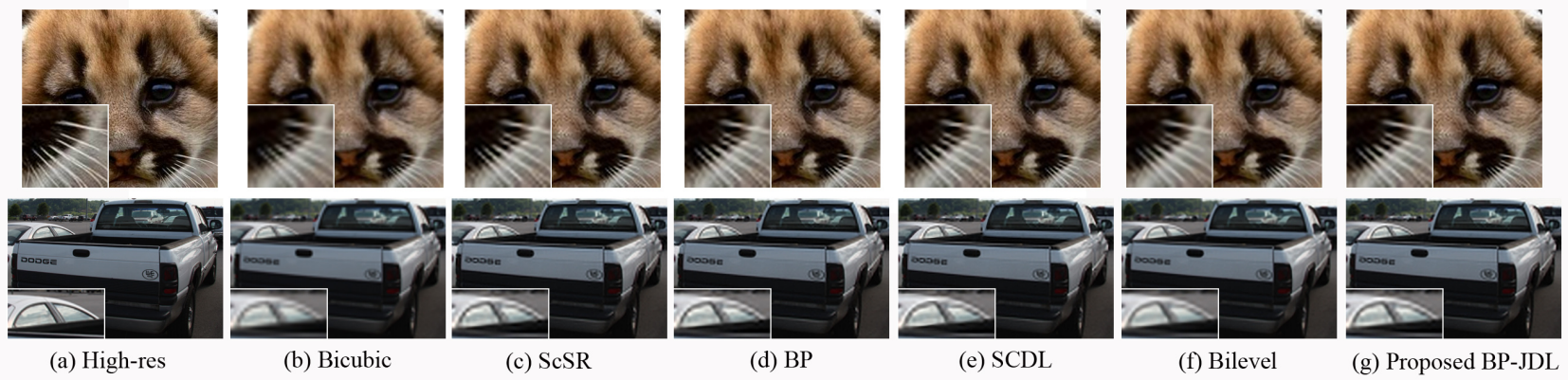



**Figure 5.16:** Visual comparison of factor of 3 super-resolution results. The upper row shows the SR results of the image Lion. The lower row shows the SR results of the image Car.

# Chapter 6

# Application of Inverse Halftoning

Halftoned image is generated by converting a grayscale continuous-tone image into a binary one that looks perceptually similar to the original one. One classical algorithm used to generate halftoned images is Floyd-Steinberg algorithm (Floyd and Steinberg, 1976). Restoring these binary images to continuous-tone ones is called "inverse halftoning". The standard error diffusion halftoning algorithm using Floyd-Steinberg filter is as follows (Floyd and Steinberg, 1976):

- Inputs: grayscale image $\mathbf{X}$ with range $[0, 1]$, filter $h_{FS}$.
- Outputs: halftoned image $\mathbf{Y}$.
- Initialization: $\tilde{\mathbf{X}}$=$\mathbf{X}$.
- For each pixel $y(i, j) \in \mathbf{Y}$, do the following three steps:
  1. $y(i, j) = 1$ if $\tilde{x}(i, j) > 0.5$. $y(i, j) = 0$ otherwise.
  2. Compute error $err = \tilde{x}(i, j) - y(i, j)$.
  3. Update $\tilde{x}(i, j)$ by distributing $e$ among unprocessed neighbors of $(i, j)$:

$$\tilde{x}(i + \delta, j + \delta) = \tilde{x}(i + \delta, j + \delta) + err \cdot h_{FS} \tag{6.1}$$

**(a)** Grayscale **(b)** Halftoned

**Figure 6.1:** Halftoned Lena image using Floyd-Steinberg. Although the halftoned image (b) looks like a grayscale image, its actually a binary image.

where Floyd-Steinberg filter is defined as

$$h_{FS} = \frac{1}{16}\begin{bmatrix} 0 & 0 & 7 \\ 1 & 5 & 1 \end{bmatrix} \tag{6.2}$$

In this application, we use the Floyd-Steinberg algorithm to generate the halftoned image and using the beta process based dictionary learning algorithm to restored the grayscale image. Example halftoned image using the Floyd-Steinberg algorithm is shown in Figure 6.1. We use 24 high-quality images from Kodak PhotoCD (Kod, 2012) for training and 5 classic images (babara, house, lena, mandrill, peppers) for testing.

## 6.1 Beta Process Dictionary Learning for Single Feature Space

We first evaluate the performance of the beta process dictionary learning for single feature space (BP) introduced in Chapter 3 for inverse halftoning (IH) from the quality of the dictionary generated as well as the fidelity of the restored image. We compare the BP with three state-of-the-art inverse halftoning algorithm: WInHD (Neelamani et al., 2002), LPA-ICI (Foi et al., 2004) and $\ell^1$ dictionary learning (L1) based algorithm (Mairal et al., 2012). Among these algorithm, WInHD and LPA-ICI is deconvolution based algorithm and L1 is sparse representation based algorithm.

We use the framework described in (Mairal et al., 2012) and use coupled dictionary learning strategy for sparse representation based algorithm. We use the grayscale patch $x$ and halftoned patch $y$ to construct the learning samples $\mathbf{v} = [x; y]$. Then we learn the dictionary based on model

$$\mathbf{v}_i = \mathbf{D}\alpha_i + \epsilon_i \tag{6.3}$$

where $\mathbf{D} = [\mathbf{D}^{(x)}; \mathbf{D}^{(y)}]$. We can use the BP or L1 to learn the dictionary. After learned the dictionary, similar to single image super-resolution reconstruction, we can use the below Algorithm 3 to reconstruct a grayscale image from a halftoned image. We extract all patches with overlaps from a test image and restore each patch independently so we get different estimates for each pixel. The estimates are then averaged to reconstruct the full image. The final image is pose-processed using BM3D (Dabov et al., 2007b) to remove possible artifacts.

For the dictionary learning details, we use 100,000 patch pairs sampled from the Kodak database. Comparing to 9 million patches used in (Mairal et al., 2012), we only use 1% of samples that used in (Mairal et al., 2012). We rescale each patch to range of $[0, 1]$ and remove the non-information patches using a threshold on the dominated measure described in Eq. 5.9, similar to what we used in SISR application. However,

**Algorithm 3** Inverse halftoning with BP dictionaries

**Input:** Halftoned image $\mathbf{Y}$, learned $\mathbf{D}^{(y)}$ and $\mathbf{D}^{(x)}$.
**Output:** Grayscale image $\mathbf{X}^*$
**Step 1** Sample patch $y_i$ from the input image $\mathbf{Y}$ with overlap $\omega$. Learn $\alpha_i$ using the L1:

$$\alpha_i = \arg\min_{\alpha_i} \frac{1}{2}\|\mathbf{D}^{(y)}\alpha_i - \mathbf{y}_i\|_2^2 + \lambda\|\alpha_i\|_1 \tag{6.4}$$

**Step 2** Recover the grayscale patch $x_i$ using $\alpha$ and learned $\mathbf{D}^{(x)}$:

$$x_i = \mathbf{D}^{(x)}\alpha_i \tag{6.5}$$

After the recovery of all grayscale patches, the initial grayscale image $\mathbf{X}_0$ can be reconstructed with overlap $\omega$.
**Step 4** The block matching 3D transfer-domain collaborative filtering (BM3D) algorithm (Dabov et al., 2007b) is used to remove artifacts and generate output grayscale image $\mathbf{X}^*$.

different to SISR application, we do not subtract mean and normalize each patch for this application, instead we just use the patch directly for the training. In SISR application, the low-res patch and high-res patch share similar mean values, what we only concern is to reconstruct the high frequency details of the images, therefore we subtract mean and normalize each patch for training to learn dictionaries that recover better details. However, in the inverse-halftoning case, the grayscale and halftone patch do not share the similar mean value and noise model, the learned dictionaries need to recover details as well as the DC level (mean value) of the patch. Therefore, we need to use the patch pair directly for training to learn a dictionaries with both capabilities. From this point of view, we can see that dictionaries learned for inverse halftoning contains more non-linear mapping compared to SISR dictionaries and hence its a more challenge task for sparse representation based algorithm.

For the BP parameters, we initial the algorithm with 512 size random generated dictionary. Similar to SISR application, since the halftone noise is not exactly Gaussian, we use a pre-defined $\sigma$ for noise variance instead of using a hyper-parameter. We tested $\sigma = r*$(data variance) with value $r = 5\%$ to $50\%$ with $5\%$ step. For the

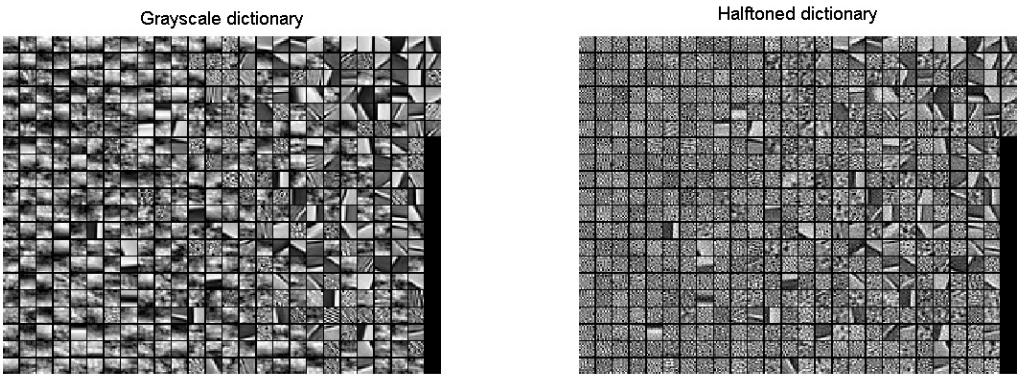



**Figure 6.2:** Learned size 506 coupled dictionary for inverse halftoning using BP. Dictionary atoms are normalized for display purpose.

patch size, we tested on $6 \times 6, 8 \times 8, 10 \times 10$ and $12 \times 12$ size patches. The best parameters found are $r = 25\%$ and patch size $10 \times 10$. We use 3000 Gibbs samples of the BP, the burn-in is 2500 samples and the dictionary is averaged using the rest 500 samples. For the L1 algorithm, we use 512 size dictionary, 2000 iterations and $\lambda = 0.05$ as found in (Mairal et al., 2012).

Dictionaries learned by BP is shown in Figure 6.2. The BP learned a 506-size dictionary successfully inferred the dictionary size non-parametrically. For the halftone dictionary, we can see that there are many noise-like dictionary atoms learned to reconstruct halftoned patches.

Compared to the SISR dictionaries learned by BP, the inverse-halftoning BP dictionary has larger sparsity. As shown in Figure 6.3, the final sparsity of the BP algorithm is $123.2/506 = 24.3\%$, whereas for the the final sparsity of BP in SISR is around 1.5%. This is because the binary halftone patches are not from natural images and need combination of many dictionary atoms to approximate. We observe the similar results for the L1 learning algorithm as well. The sparsity of L1 results is $101/512 = 19.7\%$.

For both BP and L1 learned dictionaries, we use the same reconstruction Algorithm 3 to reconstruct grayscale image. We test the $\lambda$ on the logarithmic scale

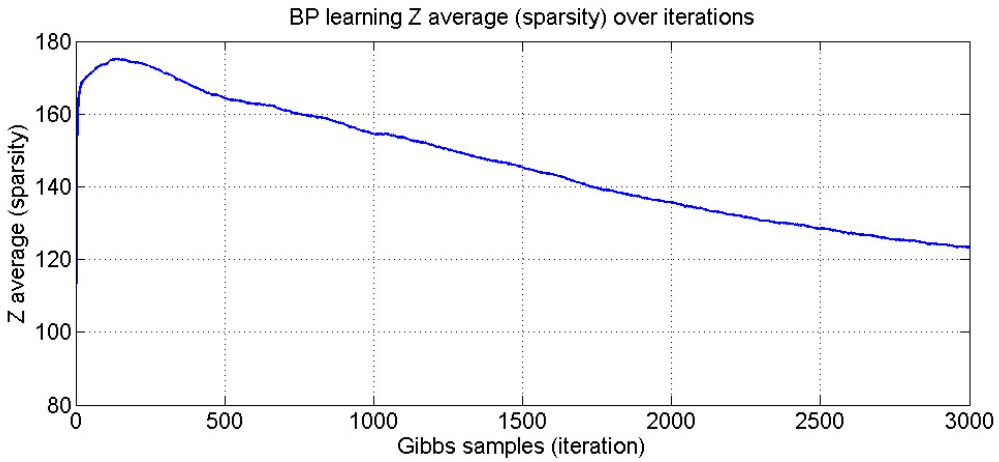



**Figure 6.3:** BP learning sparsity over iterations.

$10^i, i = -3, -2, \ldots, 0$, we found that $\lambda = 0.01$ generate the best results. We also tested on the overlap and found that max overlap (patch size - 1 = 9) generate the best results.

The inverse-halftoning results are shown in Table 6.1. We compared BP with LPA-ICI (Foi et al., 2004), WInHD (Neelamani et al., 2002) and L1 (Mairal et al., 2012). We evaluate the image quality using stat-of-the-art image quality assessment algorithms, including peak signal-to-noise ratio (PSNR), structural similarity (SSIM) Wang et al. (2004), visual information fidelity (VIF) (Sheikh and Bovik, 2006) and gradient similarity (GSM) (Liu et al., 2012). For these four evaluation methods, higher values indicate better results.

Overall, the BP results are the best among four state-of-the-art inverse-halftoning methods. Although L1 is also the sparse representation based method, its not as good as the other two convolution based methods (LPA-ICI and WInHD). This may because the 100,000 training patches is not enough. However, using 100,000 training patches BP is able to generate the best results, indicating that BP is able to capture the complex non-linear relationship between grayscale and halftoned patches. In addition, this results also prove that sparse representation based algorithms can recover both mean value and details of the grayscale image from a halftoned image.

**Table 6.1:** Comparison of inverse halftoning results.

| Image | Measures | LPA-ICI | WInHD | L1 | proposed BP |
|---|---|---|---|---|---|
| Barbara | PSNR(dB) | 25.6254 | 25.6695 | 24.5852 | **26.4029** |
| | SSIM | 0.8835 | 0.8837 | 0.8634 | **0.9034** |
| | VIF | 2.7249 | 2.6754 | 2.7790 | **2.8861** |
| | GSM | 0.9946 | 0.9945 | 0.9935 | **0.9948** |
| House | PSNR(dB) | 32.6397 | 35.2158 | 34.3150 | **35.9435** |
| | SSIM | 0.9389 | 0.9461 | **0.9521** | 0.9413 |
| | VIF | **2.4209** | 2.1353 | 2.3783 | 2.3526 |
| | GSM | 0.9940 | 0.9950 | 0.9945 | **0.9951** |
| Lena | PSNR(dB) | 32.524 | 31.8987 | 31.1593 | **32.6751** |
| | SSIM | 0.9481 | 0.9462 | 0.9461 | **0.9482** |
| | VIF | 2.4851 | 2.5219 | 2.5369 | **2.5850** |
| | GSM | 0.9963 | 0.9961 | 0.9959 | **0.9966** |
| Mandrill | PSNR(dB) | 28.2627 | 27.4198 | 26.8706 | **28.4300** |
| | SSIM | 0.9083 | 0.8834 | 0.8808 | **0.9084** |
| | VIF | 3.5185 | 3.2372 | 3.3733 | **3.5684** |
| | GSM | 0.9942 | 0.993 | 0.9912 | **0.9943** |
| Peppers | PSNR(dB) | 31.7751 | 31.0357 | 31.0802 | **32.013** |
| | SSIM | 0.9490 | **0.9499** | **0.9499** | 0.9406 |
| | VIF | 2.5266 | 2.5053 | 2.5662 | **2.5701** |
| | GSM | 0.9961 | 0.9962 | 0.996 | **0.9964** |

The visual comparison of Barbara and Lena image results are shown in Figure 6.5 and 6.6, respectively.

## 6.2 Beta Process Joint Dictionary Learning

Next, we evaluate the proposed BP-JDL algorithm on the inverse halftoning task. We directly use grayscale patch for $\mathcal{X}$ feature space and halftoned patch for $\mathcal{Y}$ feature space. Similar to BP algorithm, we use 100,000 image patches sampled for training and 5 classic image for testing. We remove the non-information patches using a threshold on the dominated measure described in Eq. 5.9. We initial the BP-JDL with 512 size random generated dictionary, and use the error variance equal to 0.25∗(data variance) for both feature space $\mathcal{X}$ and $\mathcal{Y}$ based on the test range of 5% to 50%. We also use the patch size $10 \times 10$. We use 1000 Gibbs samples of the BP-JDL, the burn-in is 950 samples and the dictionary is averaged using the rest 50 samples.

We use the Algorithm 4 to reconstruct the grayscale image from an input halftoned image. Same as the BP reconstruct algorithm, we use $\lambda = 0.01$ for the L1 reconstruction algorithm.

The BP-JDL algorithm successfully inferred 507-size dictionary with initial size of 512. The BP-JDL is able to learn dictionaries with sparsity=$34.3/507 = 6.8\%$, which is smaller than BP (24.3%) and L1 (19.7%) results. This indicates that BP-JDL is able to learn a dictionary that provide sparser solution.

For the inverse halftoning results, we found an issue with BP-JDL algorithm. The reconstructed grayscale image may have values that not between $[0, 1]$, as shown in Figure 6.4(b). This may because the relaxation the BP-JDL provided for BP cannot recover the mean value of the patch stably. For the training halftoned and grayscale patch pair, both patches have values range in $[0, 1]$, however, they don't have same mean value. The BP-JDL is not able to recover correct grayscale mean value from halftoned input patch. This may because the unimodal noise assumption of BP-JDL in halftoned spaces. As shown in Figure 6.4(a), since the halftoned image is

**Algorithm 4** Inverse halftoning with BP-JDL dictionaries

**Input:** Halftoned image $\mathbf{Y}$, learned $\mathbf{D}^{(y)}$, $\mathbf{D}^{(x)}$ and $\mathbf{M}$.
**Output:** Grayscale image $\mathbf{X}^*$
**Step 1** Sample patch $y_i$ from the input image $\mathbf{Y}$ with overlap $\omega$. Learn $\alpha_i$ using the L1:

$$\alpha_i^{(y)} = \arg\min_{\alpha_i^{(y)}} \frac{1}{2}\|\mathbf{D}^{(y)}\alpha_i^{(y)} - \mathbf{y}_i\|_2^2 + \lambda\|\alpha_i^{(y)}\|_1 \tag{6.6}$$

**Step 2** Map the sparse coefficients $\alpha^{(y)}$ to $\alpha^{(x)}$ using the learned $\mathbf{M}$:

$$\alpha_i^{(x)} = \mathbf{M}\mathbf{z}_i\alpha_i^{(y)} \tag{6.7}$$

where $\mathbf{z}_i$ is a binary vector that $z_{ik} = 1$ if $\alpha_{ik}^{(y)} \neq 0$.
**Step 3** Recover the grayscale patch $x_i$ using $\alpha$ and learned $\mathbf{D}^{(x)}$:

$$x_i = \mathbf{D}^{(x)}\alpha_i^{(x)} \tag{6.8}$$

After the recovery of all grayscale patches, the initial grayscale image $\mathbf{X}_0$ can be reconstructed with overlap $\omega$.
**Step 5** The block matching 3D transfer-domain collaborative filtering (BM3D) algorithm (Dabov et al., 2007b) is used to remove artifacts and generate output grayscale image $\mathbf{X}^*$.

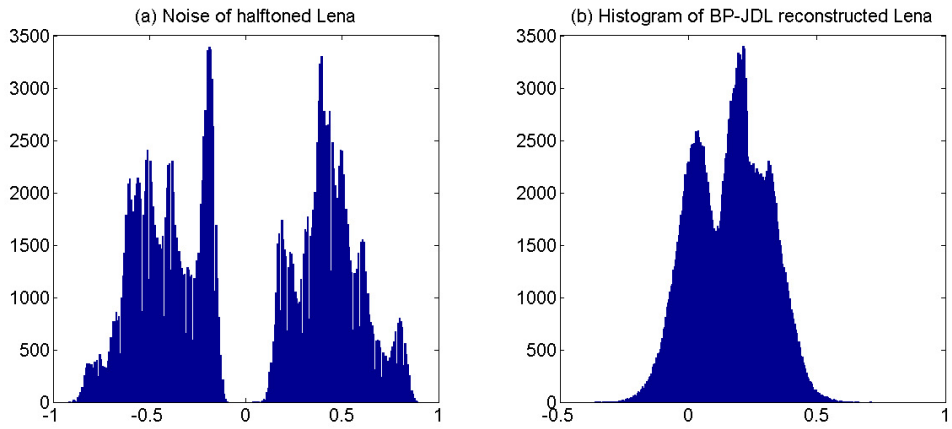



**Figure 6.4:** (a)Noise histogram of (halftoned lena - lena), (b) Histogram of BP-JDL reconstructed lena, the pixel values are not between $[0, 1]$.

binary, if we subtract the grayscale lena from halftoned lena, we can see the noise is bimodal not unimodal. In the BP-JDL model described in Chapter 4, we assume unimodal Gaussian noise for both feature spaces. Therefore, in the inverse halftoning application, the BP-JDL is trying to model bimodal noise of halftoned space using unimodal noise model. This may be the reason of BP-JDL cannot stably recover the DC level (mean value) of grayscale patch.

However, BP-JDL is still able to recover the high frequency details of grayscale image from halftoned image. Because pixel values of BP-JDL restored image is not between $[0, 1]$, we cannot compare the recover accuracy using the PSNR, SSIM etc. evaluation metrics. However, we can compare the BP-JDL results visually by normalizing the image to the range of $[0, 1]$. After the normalization, we can visually compare the results of BP-JDL with other state-of-the-art inverse halftoning results.

The inverse halftoning results of barbara and lena are shown in Figure 6.5 and 6.6, respectively. From the visual comparison results, we can clearly see that BP-JDL is able to provide the most details compared to other four algorithms. The L1 results has least artifacts, but its overly smooth and recover least details compared to other algorithms. The LPA-ICI results is noisy than other results. The WInHD can provide some level of details, however, these details are fragmented. The BP is able to provide

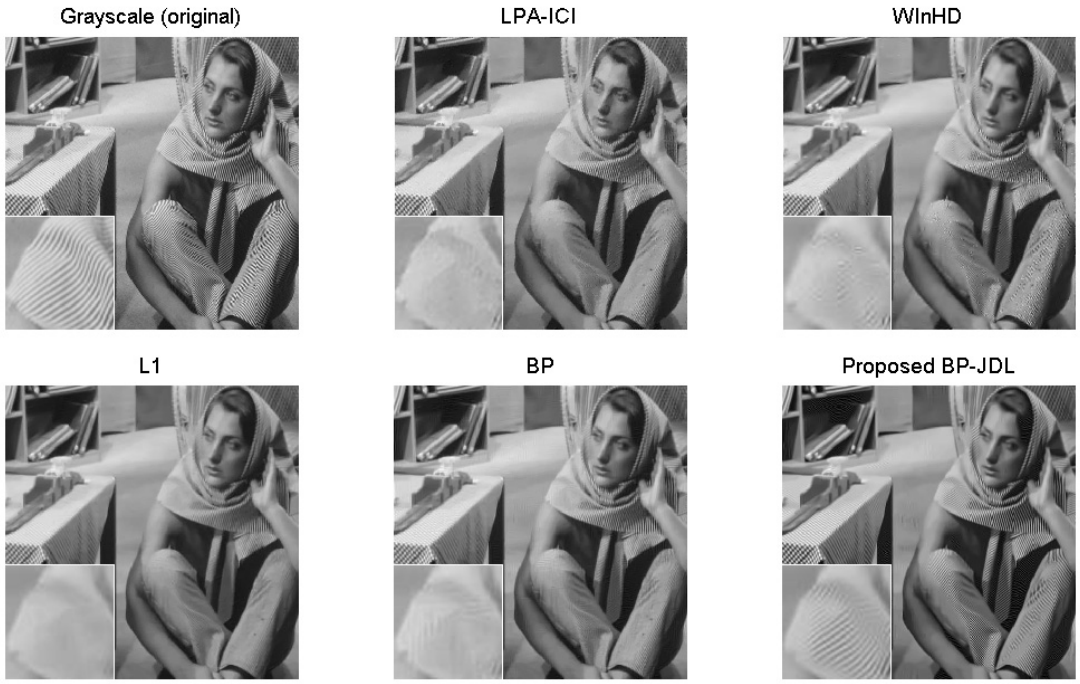



**Figure 6.5:** Inverse halftoning results comparison of Barbara.

a smooth and rich detailed image, however, it has some blocky artifacts from the reconstruction. Although the BP-JDL introduced small artifacts in homogeneous regions, it provides the best recover details.

From the inverse halftoning applications, we can see that the BP-JDL may not suitable for applications that demonstrate bimodal noise in one of the feature spaces.

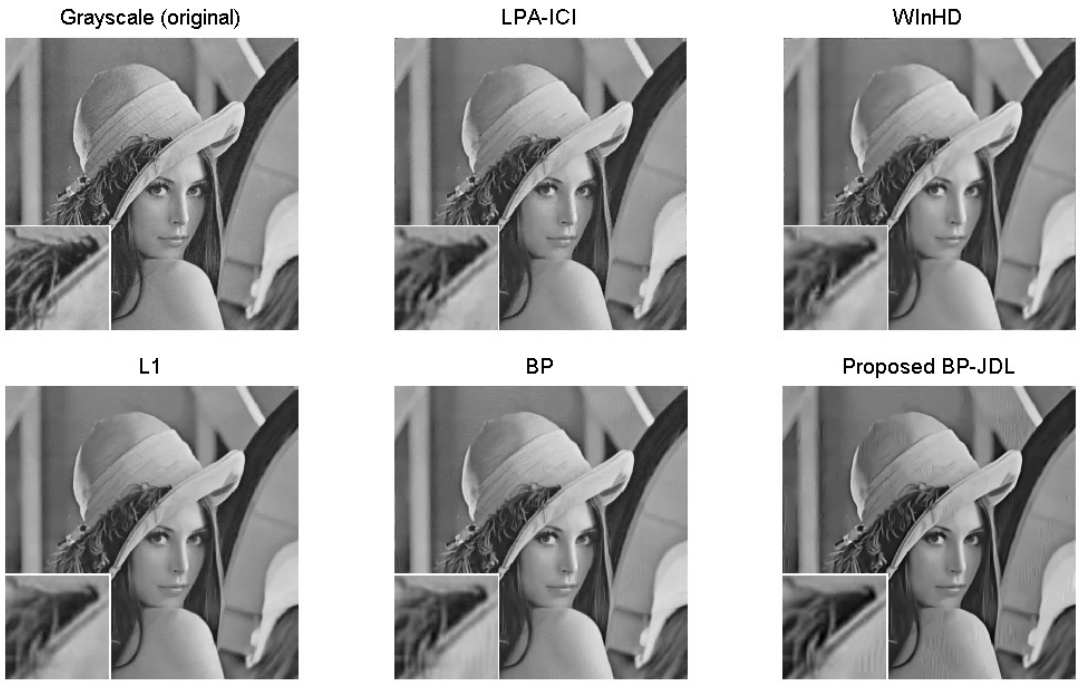



**Figure 6.6:** Inverse halftoning results comparison of Lena.

# Chapter 7

# Conclusions and Future Work

## 7.1 Summary

In this article, a Bayesian method using beta process (BP) was proposed for solving dictionary learning problem single feature space and a beta process joint dictionary learning (BP-JDL) method was proposed for solving the dictionary learning problem in coupled feature spaces. The BP was compared with two other state-of-the-art single feature space dictionary learning algorithms for the application of single image super-resolution (SISR), which a single low-resolution image was reconstructed to a high-resolution image using the learned over-complete dictionaries. The experiment results showed that BP is able to infer an appropriate dictionary size and sparsity level non-parametrically. Moreover, the SISR results of BP were the best among the three single feature space dictionary learning algorithms in terms of the recovery accuracy. Next, the BP was compared with three other state-of-the-art algorithms for the application of inverse halftoning. The experiments results showed that the learned BP dictionaries are also the best among four methods in terms of the image quality and the recovery accuracy.

We further extend the BP to the new BP-JDL algorithm that can better solve the problem of dictionary learning in coupled feature spaces. The proposed BP-JDL

method could have wide applications in the field of signal processing because many problems in this area require the mapping between two feature spaces. In order to demonstrate the capability of proposed BP-JDL algorithm, we also applied BP-JDL method to solve the SISR problem and inverse halftoning problem. For the SISR, five state-of-the-art dictionary learning based SISR methods were compared with BP-JDL in terms of the quality of dictionary generated and the quality of the super-resolution images. The experimental results showed that the BP-JDL method is able to learn dictionaries that fit the coupled feature spaces better than previous methods. The SISR results showed that the images reconstruction using BP-JDL have the best overall quality compared to other four methods. In addition, BP-JDL was able to infer an appropriate dictionary size non-parametrically as same as BP. For the inverse halftoning application, the BP-JDL algorithm is able to learn dictionaries that recover better high-frequency details than four other inverse halftoning methods, including BP. Successful applications of BP-JDL prove that BP-JDL is able to model complex non-linear relationship between two feature spaces and is an effective algorithm for dictionary learning in coupled feature spaces.

## 7.2 Future Research

In the future, we can discover more properties and applications about proposed research, which we will summarize below.

### 7.2.1 Improvement of the BP-JDL

There are several possible improvements of the proposed algorithm:

#### Inference

Similar to the variational Bayesian inference provided for the BP model (Paisley and Carin, 2009; Chen et al., 2010), a variational Bayesian inference could be used for the

BP-JDL inference, which may have a faster convergence speed than Gibbs sampler. In addition, (Miller, 2011) pointed out that Gibbs sampling may converge quickly but stuck in widely varying local optima while variational inference is slower per iteration but find regions of higher predictive likelihood. We derived the the inference equation which can be found in Appendix A.

### Integration of the mapping matrix

The mapping matrix $\mathbf{M}$ of BP-JDL is solved via least square. However, if we integrate the mapping matrix to the model, we may learn a better mapping matrix and we may accelerate the learning speed since we don't need to sample two coefficients matrix. With the integration of the $\mathbf{M}$ matrix, the model change be modified as:

$$
\begin{aligned}
\mathbf{x}_i = \mathbf{D}^{(x)}\alpha_i^{(x)} + \epsilon_i^{(x)}, \quad & \mathbf{y}_i = \mathbf{D}^{(y)}\alpha_i^{(y)} + \epsilon_i^{(y)} \\
\alpha_i^{(x)} = \mathbf{z}_i \circ \mathbf{M}\mathbf{s}_i^{(y)}, \quad & \alpha_i^{(y)} = \mathbf{z}_i \circ \mathbf{s}_i^{(y)} \\
\mathbf{d}_k^{(x)} \sim \mathcal{N}(0, P_x^{-1}\mathbf{I}_{P_x}), \quad & \mathbf{d}_k^{(y)} \sim \mathcal{N}(0, P_y^{-1}\mathbf{I}_{P_y}) \\
\mathbf{m}_{jk} \sim \mathcal{N}(0, \gamma_M^{-1}\mathbf{I}_K), \quad & \mathbf{s}_i^{(y)} \sim \mathcal{N}(0, \gamma_{s^{(y)}}^{-1}\mathbf{I}_K) \\
\mathbf{z}_i \sim \prod_{k=1}^{K} Bernoulli(\pi_k), \quad & \pi_k \sim Beta(a/K, b(K-1)/K) \\
\epsilon_i^{(x)} \sim \mathcal{N}(0, \gamma_{\epsilon^{(x)}}^{-1}\mathbf{I}_{P_x}), \quad & \epsilon_i^{(y)} \sim \mathcal{N}(0, \gamma_{\epsilon^{(y)}}^{-1}\mathbf{I}_{P_y}) \\
\gamma_{s^{(y)}} \sim \Gamma(c,d), \gamma_M \sim \Gamma(g,h), \quad & \gamma_{\epsilon^{(x)}}, \gamma_{\epsilon^{(y)}} \sim \Gamma(e,f)
\end{aligned}
\tag{7.1}
$$

The difference between this model and BP-JDL mode in Eq. 4.3 is the $\mathbf{s}_i^{(x)}$ is replaced with $\mathbf{M}\mathbf{s}_i^{(y)}$. The mapping matrix $\mathbf{M}$ is integrated into the model and no longer calculated after the dictionary learning process. We could assume that each element in $\mathbf{M}$ is normal distributed as $\mathbf{m}_{jk} \sim \mathcal{N}(0, \gamma_M^{-1}\mathbf{I}_K)$, and $\gamma_M$ is a Gamma distributed hyper parameter. The model is fully conjugated and we can also use the Gibbs sampler or Variational method for inference.

### 7.2.2 Evaluation

- Currently, the quality of dictionary is evaluated by the recovery accuracy of super-resolution reconstruction. There is no direct evaluation for the quality of the dictionary. According to the receptive fields properties (Olshausen and Fieldt, 1996), we may can evaluate the quality of the dictionary by evaluating the localized, oriented and bandpass properties of patches in the learned dictionaries.

- The objective metric (PSNR, SSIM, VIF and GSM) was used for evaluation of the result images. However, we may use other image quality assessment metric such as the metric that do not need the reference image (He et al., 2012). We may also use a subjective method to evaluate the quality of the image.

### 7.2.3 Other Applications

The BP-JDL may have wide applications. Except for the applications discussed in this article, the BP-JDL algorithm could be used in other inverse problem such as intrinsic image estimation, digital art authentication etc. In addition, the algorithm can be used in applications that require to learn dictionary in coupled feature spaces, such as invariant human pose estimation etc.

For the sketch-photo applications, although (Wang et al., 2012) showed experimented result based on SCDL, the patch-based approach may not suitable for this application. For instance, the training dataset such as CUFS (Tang and Wang, 2003) contain photo and hand draw sketches. At the low level ($7 \times 7$ size patch), the alignment of two patches in sketch and photo may largely different and the dictionaries learned may not be effective. This is also the reason for blurry results shown in (Wang et al., 2012).

# Bibliography

(2012). Kodak photocd dataset. http://r0k.us/grahpics/kodak/. 74

Aharon, M., Elad, M., and Bruckstein, A. (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transcations on Signal Processing*, 54(11). 2, 3, 5, 12

Aldous, D. (1985). Exchangeability and related topics. In Hennequin, P., editor, *cole d't de Probabilits de Saint-Flour XIII ? 1983*, volume 1117 of *Lecture Notes in Mathematics*, pages 1–198. Springer Berlin Heidelberg. 16

Beal, M. (2003). *Variational algorithms for approximate bayesian inference*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, London, UK. 102

Billingsley, P. (1995). *Probability and measure*. Wiley Press, New York, 3 edition. 14

Buades, A., Coll, B., and Morel, J.-M. (2005). A non-local algorithm for image denoising. In *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 60 – 65 vol. 2. 26, 61, 62

Capel, D. (2001). Image mosaicing and super-resolution. *Ph.D. Thesis, University of Oxford*. 62

Chandler, D. and Hemami, S. (2007). Vsnr: A wavelet-based visual signal-to-noise ratio for natural images. *IEEE Transactions on Image Processing*, 16(9):2284–2298. 30

Chang, H., Yeung, D.-Y., and Xiong, Y. (2004). Super-resolution through neighbor embedding. In *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1. 2

Chen, B., Chen, M., Paisley, J., Zaas, A., Woods, C., Ginsburg, G., Hero, Alfred, I., Lucas, J., Dunson, D., and Carin, L. (2010). Bayesian inference of the number of factors in gene-expression analysis: application to human virus challenge studies. *BMC Bioinformatics*, 11:1–16. 86

Chen, G.-H., Yang, C.-L., Po, L.-M., and Xie, S.-L. (2006). Edge-based structural similarity for image quality assessment. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 2, pages II–II. 31, 32

Chen, S. S., Donoho, D. L., Michael, and Saunders, A. (1998). Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61. 2

Chung, K.-L. and Wu, S.-T. (2005). Inverse halftoning algorithm using edge-based lookup table approach. *IEEE Transactions on Image Processing*, 14(10):1583–1589. 27

Colson, B., Marcotte, P., and Savard, G. (2007). An overview of bilevel optimization. *Annals of Operations Research*, pages 235 –256. 23

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. (2007a). Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. on Image Processing*, 16(8):2080 –2095. 26, 62

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. (2007b). Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095. 75, 76, 81

Donoho, D. L. (2006). For most large underdetermined systems of linear equations the minimal $\ell^1$-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics*, 59(6). 10

Edelman, A. (1988). Eigenvalues and condition numbers of random matrices. *SIAM J. Matrix Anal. Appl.*, 9(4):543–560. 63

Efros, A. A. and Freeman, W. T. (2001). Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 341–346. ACM. 2

Engan, K., Aase, S., and Hakon Husoy, J. (1999). Method of optimal directions for frame design. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 5, pages 2443 –2446 vol.5. 2

Farsiu, S., Robinson, M., Elad, M., and Milanfar, P. (2004). Fast and robust multiframe super resolution. *IEEE Trans. on Image Processing*, 13(10). 24

Ferguson, T. S. (1973). A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230. 15

Floyd, R. W. and Steinberg, L. (1976). An Adaptive Algorithm for Spatial Greyscale. *Proceedings of the Society for Information Display*, 17(2):75–77. 73

Foi, A., Katkovnik, V., Egiazarian, K., and Astola, J. (2004). Inverse halftoning based on the anisotropic lpa-ici deconvolution. In *PROCEEDINGS OF THE 2004 INTERNATIONAL TICSP WORKSHOP ON SPECTRAL METHODS AND MULTIRATE SIGNAL PROCESSING, SMMSP 2004*, pages 49–56. 28, 75, 78

Fristedt, B. and Gray, L. (1997). *A Modern Approach to Probability Theory*. Birkhauser, Boston, MA. 13, 14

Gao, X., Lu, W., Tao, D., and Li, X. (2009). Image quality assessment based on multiscale geometric analysis. *IEEE Transactions on Image Processing*, 18(7):1409–1423. 30

Gorodnitsky, I., George, J., and Rao, B. (1995). Neuromagnetic source imaging with FOCUSS: a recursive weighted minimum norm algorithm. *Electroencephalography and Clinical Neurophysiology*, 95(4):231–251. 12

Griffiths, T. L. and Ghahramani, Z. (2005). Infinite latent feature models and the indian buffet process. In *Proceeding of Advances in Neural Information Processing Systems (NIPS)*. 4, 5

Hancock, P. J. B., Baddeley, R. J., and Smith, L. S. (1991). The principal components of natural images. *Network: Comput.Neural Syst.*, 3:61–72. 1

He, L., Tao, D., Li, X., and Gao, X. (2012). Sparse representation for blind image quality assessment. In *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1146 –1153. 88

Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B., and Salesin, D. H. (2001). Image analogies. In *Proc. of SIGGRAPH*, pages 327–340. 2, 3

Hjort, N. (1990). Nonparametric bayes estimators based on beta processes in models for life history data. *annals of statistics*, (1):1259–1294. 13

Ho, K.-C. (2004). Iterated conditional modes for inverse halftoning. In *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, volume 3, pages III–901–III–904 Vol.3. 27

Jia, K., Wang, X., and Tang, X. (2013). Image transformation based on learning dictionaries across image spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):367–380. 2

Kim, D.-O., Han, H.-S., and Park, R.-H. (2010). Gradient information-based image quality metric. *Consumer Electronics, IEEE Transactions on*, 56(2):930–936. 31, 32

Kite, T., Damera-Venkata, N., Evans, B., and Bovik, A. (2000). A fast, high-quality inverse halftoning algorithm for error diffused halftones. *IEEE Transactions on Image Processing*, 9(9):1583–1592. 27

Knowles, D. and Ghahramani, Z. (2007). Infinite sparse factor analysis and infinite independent components analysis. In *Independent Component Analysis and Signal Separation*, volume 4666. 5

Kreutz-Delgado, K., Murray, J. F., Rao, B. D., Engan, K., Lee, T.-W., and Sejnowski, T. J. (2003). Dictionary learning algorithms for sparse representation. *Neural Computing*, 15(2):349–396. 12

Kreutz-Delgado, K. and Rao, B. D. (2002). Focuss-based dictionary learning algorithms. volume 4119-53. 2

Larson, E. C. and Chandler, D. M. (2010). Most apparent distortion: full-reference image quality assessment and the role of strategy. *Journal of Electronic Imaging*, 19(1):011006–011006–21. 30

Lee, H., Battle, A., Raina, R., and Y. Ng, A. (2007). Efficient sparse coding algorithms. In *Proceeding of Advances in Neural Information Processing Systems (NIPS)*. 2, 10, 61

Lei, Z. and Li, S. (2009). Coupled spectral regression for matching heterogeneous faces. In *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1123 –1128. 2, 3

Lewicki, M. S., Sejnowski, T. J., and Hughes, H. (1998). Learning overcomplete representations. *Neural Computation*, 12:337–365. 2

Lin, D. and Tang, X. (2005). Coupled space learning of image style transformation. In *Proc. of ICCV*, volume 2, pages 1699 –1706 Vol. 2. 2, 3, 22

Liu, A., Lin, W., and Narwaria, M. (2012). Image quality assessment based on gradient similarity. *IEEE Transactions on Image Processing*, 21(4):1500–1512. 28, 30, 31, 34, 68, 78

Lu, X., Yuan, H., Yan, P., Yuan, Y., and Li, X. (2012). Geometry constrained sparse coding for single image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1648–1655. 24

Mairal, J., Bach, F., and Ponce, J. (2012). Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):791–804. 2, 11, 28, 75, 77, 78

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2009a). Online dictionary learning for sparse coding. In *Proc. of the ICML*. 2, 11

Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A. (2009b). Non-local sparse models for image restoration. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2272 –2279. 26

Mallat, S. and Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transcations on Signal Processing*, 41(12):3397 –3415. 1

Miller, K. T. (2011). *Bayesian Nonparametric Latent Feature Models*. PhD thesis, University of California, Berkeley, Berkeley, CA. 13, 15, 87

Murray, J. F. and Kreutz-Delgado, K. (2007). Learning sparse overcomplete codes for images. *J. VLSI Signal Process. Syst.*, 46. 2

N. Ponomarenko, F. Silvestri, K. E. M. C. J. A. and Lukin, V. (2007). On between-coefficient contrast masking of dct basis functions. In *Proc. 3rd Int. Workshop Video Process. Qual. Metrics Consum. Electron.* 30

Neelamani, R., Nowak, R., and Baraniuk, R. (2002). Winlid: Wavelet-based inverse halftoning via deconvolution. 27, 75, 78

Olshausen, B. A. and Fieldt, D. J. (1996). Natural image statistics and efficient coding. *Network Bristol England*, 7(2). 1, 63, 88

Olshausen, B. A. and Fieldt, D. J. (1997). Sparse coding with an overcomplete basis set: a strategy employed by v1. *Vision Research*, 37. 1

Ong, E., Lin, W., Lu, Z., Yao, S., and Etoh, M. (2004). Visual distortion assessment with emphasis on spatially transitional regions. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(4):559–566. 31

Paisley, J. and Carin, L. (2009). Nonparametric factor analysis with beta process priors. In *Proc. of ICML*. 2, 4, 5, 6, 7, 13, 15, 16, 17, 18, 36, 38, 41, 86, 102

Rai, P. and Daumé III, H. (2008). The infinite hierarchical factor regression model. In *Proceeding of Advances in Neural Information Processing Systems (NIPS)*. 5

Ran, X. and Farvardin, N. (1995). A perceptually motivated three-component image model-part i: description of the model. *IEEE Transactions on Image Processing*, 4(4):401–415. 31

Ranzato, M., Poultney, C., Chopra, S., and LeCun, Y. (2006). Efficient learning of sparse representations with an energy-based model. In *Proceeding of Advances in Neural Information Processing Systems (NIPS)*. 2

Rao, B. D., Member, S., Kreutz-delgado, K., and Member, S. (1999). An affine scaling methodology for best basis selection. *IEEE Transcations on Signal Processing*, pages 187–200. 12

Sharma, A. and Jacobs, D. W. (2011). Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch. In *Proceedings of the 2011 IEEE Conference*

*on Computer Vision and Pattern Recognition*, pages 593–600, Washington, DC, USA. IEEE Computer Society. 2

Sheikh, H. and Bovik, A. (2006). Image information and visual quality. *IEEE Transactions on Image Processing*, 15(2):430–444. 30, 32, 34, 66, 78

Son, C.-H. (2012). Inverse halftoning based on sparse representation. *Opt. Lett.*, 37(12):2352–2354. 2, 28

Son, C.-H. and Choo, H. (2013). Iterative inverse halftoning based on texture-enhancing deconvolution and error-compensating feedback. *Signal Processing*, 93(5):1126 – 1140. 27

Stevenson, R. L. (1997). Inverse halftoning via map estimation. *IEEE Transactions on Image Processing*, 6(4):574–583. 27

Sun, J. and Tappen, M. F. (2011). Learning non-local range markov random field for image restoration. In *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2745–2752. 26

Sun, J., Zheng, N.-N., Tao, H., and Shum, H.-Y. (2003). Image hallucination with primal sketch priors. In *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2. 2

Tang, X. and Wang, X. (2003). Face sketch synthesis and recognition. In *Proc. of ICCV*, pages 687 –694 vol.1. 2, 88

Thibaux, R. and Jordan, M. I. (2007). Hierarchical beta processes and the indian buffet process. In *Proc. of International Conference on Artificial Intelligence and Statistics*. 14, 16

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58(1). 2

Tipping, M. E. (2001). Sparse bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1. 2

Tipping, M. E. and Bishop, C. M. (2003). Bayesian image super-resolution. In *Proceeding of Advances in Neural Information Processing Systems (NIPS)*. MIT Press. 24

TSUTOMU, T., HIDETO, N., MASAHIRO, S., and NOBUHIKO, S. (1999). Development of new driving method for ac-pdps. In *Proc. of the Sixth International Display Workshops*. 27

Vaidyanathan, P. P. (2001). Look-up table (lut) method for inverse halftoning. *IEEE Trans. Image Processing*, 10:1566–1578. 27

Wainwright, M. J., Simoncelli, E. P., and Willsky, A. S. (2001). Random cascades on wavelet trees and their use in analyzing and modeling natural images. *Applied and Computational Harmonic Analysis*, 11:89–123. 33

Wang, J., Zhu, S., and Gong, Y. (2010). Resolution enhancement based on learning the sparse association of image patches. *Pattern Recognition Letters*, 31(1):1 – 10. 24

Wang, S., Zhang, L., Liang, Y., and Pan, Q. (2012). Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2216 –2223. 2, 3, 4, 6, 21, 24, 26, 61, 63, 69, 88

Wang, X. and Tang, X. (2009). Face photo-sketch synthesis and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):1955 –1967. 2, 3

Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Trans. on Image Processing*, 13(4):600 –612. 29, 30, 31, 50, 54, 66, 78

Yang, J., Wang, Z., Lin, Z., Shu, X., and Huang, T. (2012a). Bilevel sparse coding for coupled feature spaces. In *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2360 –2367. 2, 3, 4, 6, 22, 24, 41, 61, 63, 69

Yang, J., Wright, J., Huang, T., and Ma, Y. (2008). Image super-resolution as sparse representation of raw image patches. In *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2, 3, 5, 10, 20, 24, 25, 48, 63

Yang, J., Wright, J., Huang, T., and Ma, Y. (2010). Image super-resolution via sparse representation. *IEEE Transcation on Image Processing*, 19(11). 4, 5, 41, 49, 61, 65, 66, 68

Yang, S., Wang, M., Chen, Y., and Sun, Y. (2012b). Single-image super-resolution reconstruction via learned geometric dictionaries and clustered sparse coding. *IEEE Transactions on Image Processing*, 21(9):4016–4028. 24, 63

Zeyde, R., Elad, M., and Protter, M. (2010). On single image scale-up using sparse-representation. In *Proc. of Internation Conference on Curves and Surfaces*. 2, 3, 20, 63, 68

Zhou, M., Chen, H., Paisley, J., Ren, L., Li, L., Xing, Z., Dunson, D., Sapiro, G., and Carin, L. (2012). Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images. *IEEE Transcations on Image Processing*, 21(1):130 –144. 4, 41, 42

Zhou, M., Chen, H., Paisley, J., Ren, L., Sapiro, G., and Carin, L. (2009). Non-parametric bayesian dictionary learning for sparse image representations. In *Proceeding of Advances in Neural Information Processing Systems (NIPS)*. 2, 5, 7, 8, 13, 17, 19, 36, 37, 38

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320. 2, 11

# Appendix

# A Variational Inference of Beta Process Joint Dictionary Learning

Similar to (Paisley and Carin, 2009), we derive a variational Bayesian algorithm (Beal, 2003) for fast inference of BP-JDL model of Eq. 4.3.

## A.1 The VB-E Step

- Update for $z_{ik}$

$$\begin{aligned} p(z_{ik}=1|-) \sim & \mathcal{N}(x_i; \mathbf{D}^{(x)}(z_i \circ s_i^{(x)}), \gamma_{\epsilon^{(x)}}^{-1} I_{P_x}) \\ & \mathcal{N}(y_i; \mathbf{D}^{(y)}(z_i \circ s_i^{(y)}), \gamma_{\epsilon^{(y)}}^{-1} I_{P_y}) \\ & Bernoulli(z_{ik}; \pi_k) \end{aligned} \tag{2}$$

The posterior probability of $z_{ik}=1$ can be expressed as:

$$\begin{aligned} & p(z_{ik}=1|-) \\ \propto \quad & \exp[\langle \pi_k \rangle] \exp[ -\frac{\gamma_\epsilon^{(x)}}{2}(\langle {s_{ik}^{(x)}}^2 \rangle \langle {\mathbf{d}_k^{(x)}}^T \mathbf{d}_k^{(x)} \rangle - 2\langle s_{ik}^{(x)} \rangle \langle \mathbf{d}_k^{(x)} \rangle^T \langle \mathbf{x}_i^{-k} \rangle) \\ & -\frac{\gamma_\epsilon^{(y)}}{2}(\langle {s_{ik}^{(y)}}^2 \rangle \langle {\mathbf{d}_k^{(y)}}^T \mathbf{d}_k^{(y)} \rangle - 2\langle s_{ik}^{(y)} \rangle \langle \mathbf{d}_k^{(y)} \rangle^T \langle \mathbf{y}_i^{-k} \rangle)] \end{aligned} \tag{3}$$

where $\langle \cdot \rangle$ indicates the expectation. The posterior probability of $z_{ik}=0$ can be expressed as:

$$p(z_{ik}=0|-) \propto \exp[\langle ln(1-\pi_k) \rangle] \tag{4}$$

The expectation can be calculated as

$$\begin{aligned} \langle ln(\pi_k) \rangle &= \psi(\frac{a}{K} + \sum_{i=1}^{N} \langle z_{ik} \rangle) - \psi(\frac{a+b(K-1)}{K} + N) \\ \langle ln(1-\pi_k) \rangle &= \psi(\frac{b(K-1)}{K} + N - \sum_{i=1}^{N} \langle z_{ik} \rangle) - \psi(\frac{a+b(K-1)}{K} + N) \end{aligned} \tag{5}$$

where $\psi(\cdot)$ represents teh digamma function and

$$\begin{aligned} \langle {s_{ik}^{(x)}}^2 \rangle &= \langle s_{ik}^{(x)} \rangle^2 + \Sigma^k_{s_{ik}^{(x)}} \\ \langle {\mathbf{d}_k^{(x)}}^T \mathbf{d}_k^{(x)} \rangle &= \langle \mathbf{d}_k^{(x)} \rangle^T \langle \mathbf{d}_k^{(x)} \rangle + trace(\Sigma_{\mathbf{d}_k^{(x)}}); \end{aligned} \tag{6}$$

where $\Sigma_{\mathbf{d}_k^{(x)}}$ is defined in the update for $\mathbf{d}_k$ and $\Sigma^k_{s_{ik}^{(x)}}$ is the $k^{th}$ diagonal element of $\Sigma_{s_{ik}^{(x)}}$ defined in the update for $s_{ik}^{(x)}$. The $\langle {s_{ik}^{(y)}}^2 \rangle$ and $\langle {\mathbf{d}_k^{(y)}}^T \mathbf{d}_k^{(y)} \rangle$ can be calculated in the similar way.

## A.2 The VB-M Step

- Update for $\mathbf{d}_k^{(x)}$

$$p(\mathbf{d}_k^{(x)}|-) \sim \mathcal{N}(\mathbf{d}_k^{(x)}; 0, P_x^{-1} I_{P_x}) \prod_{i=1}^{N} \mathcal{N}(x_i; \mathbf{D}^{(x)}(z_i \circ s_i^{(x)}), \gamma^{-1}_{\epsilon^{(x)}} I_{P_x}) \tag{7}$$

$\mathbf{d}_k$ can be drawn from a normal distribution

$$p(\mathbf{d}_k^{(x)}|-) \sim \mathcal{N}(\mu_{\mathbf{d}_k^{(x)}}, \Sigma_{\mathbf{d}_k^{(x)}}) \tag{8}$$

and

$$\begin{aligned} \Sigma_{\mathbf{d}_k^{(x)}} &= (P_x \mathbf{I} + \gamma_\epsilon^{(x)} \sum_{i=1}^{N} \langle z_{ik}^2 \rangle \langle {s_{ik}^{(x)}}^2 \rangle)^{-1} \\ \mu_{\mathbf{d}_k^{(x)}} &= \gamma_\epsilon^{(x)} \Sigma_{\mathbf{d}_k^{(x)}} \sum_{i=1}^{N} \langle z_{ik} \rangle \langle s_{ik}^{(x)} \rangle \langle \mathbf{x}_i^{-k} \rangle \end{aligned} \tag{9}$$

where $\mathbf{x}_i^{-k} = \mathbf{x}_i - \mathbf{D}(\mathbf{s}_i^{(x)} \circ \mathbf{z}_i) + \mathbf{d}_k^{(x)}(s_{ik}^{(x)} \circ z_{ik})$ and $\langle {s_{ik}^{(x)}}^2 \rangle$ is given in Eq 6.

- Update for $s_i^{(x)}$

$$p(\mathbf{s}_i^{(x)}|-) \sim \mathcal{N}(x_i; \mathbf{D}^{(x)}(z_i \circ s_i^{(x)}), \gamma^{-1}_{\epsilon^{(x)}} I_{P_x}) \mathcal{N}(s_i^{(x)}; 0, \gamma^{-1}_{s^{(x)}} I_K) \tag{10}$$

$\mathbf{s}_i$ can be drawn from a normal distribution

$$p(s_i^{(x)}|-) \sim \mathcal{N}(\mu_{s_i^{(x)}}, \Sigma_{s_i^{(x)}}) \tag{11}$$

and

$$\begin{aligned} \Sigma_{s_i^{(x)}} &= (\gamma_s^{(x)} + \gamma_\epsilon^{(x)} \langle \tilde{\mathbf{D}}_i^{(x)^T} \tilde{\mathbf{D}}_i^{(x)} \rangle)^{-1} \\ \mu_{s_i^{(x)}} &= \gamma_\epsilon^{(x)} \Sigma_{s_i^{(x)}} (\langle \tilde{\mathbf{D}}_i^{(x)} \rangle^T \mathbf{x}_i) \end{aligned} \tag{12}$$

where we define $\tilde{\mathbf{D}}_i^{(x)} = \mathbf{D}_i \circ \tilde{Z}_i$ and $\tilde{Z}_i = [z_i, \cdots, z_i]^T$, with the $K$-dimensional vector, $z_i$, repeated $P$ times. Given that $\langle \tilde{\mathbf{D}}_i^{(x)} \rangle = \langle \mathbf{D}_i \rangle \circ \langle \tilde{Z}_i \rangle$,we can calculate

$$\langle \tilde{\mathbf{D}}_i^{(x)^T} \tilde{\mathbf{D}}_i^{(x)} \rangle = (\langle \mathbf{D}_i^{(x)^T} \mathbf{D}_i^{(x)} \rangle + A) \circ (\langle z_i \rangle \langle z_i \rangle^T + B_i) \tag{13}$$

where $A$ and $B_i$ are calculated as follows

$$\begin{aligned} A &= diag[trace(\Sigma_{\mathbf{d}_1^{(x)}}), \cdots, trace(\Sigma_{\mathbf{d}_K^{(x)}})] \\ B &= diag[\langle z_{i1} \rangle (1 - \langle z_{i1} \rangle), \cdots, \langle z_{iK} \rangle (1 - \langle z_{iK} \rangle)] \end{aligned} \tag{14}$$

- Update for $\pi_k$

$$p(\pi_k|-) \ \sim Beta(\pi_k; a, b) \prod_{i=1}^{N} Bernoulli(z_{ik}; \pi_k) \tag{15}$$

$\pi_k$ can be drawn from a Beta distribution as

$$p(\pi_k|-) \ \sim Beta(\pi_k; a, b) \tag{16}$$

where

$$\begin{aligned} a &= \frac{a_0}{K} + \sum_{i=1}^{N} \langle z_{ik} \rangle \\ b &= \frac{b_0(K-1)}{K} + N - \sum_{i=1}^{N} \langle z_{ik} \rangle \end{aligned} \tag{17}$$

- Update for $\gamma_{s^{(x)}}$

$$p(\gamma_{s^{(x)}}|-) \sim \Gamma(\gamma_{s^{(x)}}; c, d) \prod_{i=1}^{N} \mathcal{N}(s_i^{(x)}; 0, \gamma_{s^{(x)}}^{-1} I_K) \tag{18}$$

$\gamma_{s^{(x)}}$ can be drawn from a Gamma distribution as $p(\gamma_{s^{(x)}}|-) \sim \Gamma(c', d')$, where

$$\begin{aligned} c' &= c + \frac{1}{2} KN \\ d' &= d + \frac{1}{2} \sum_{i=1}^{N} (\langle \mathbf{s}_i^{(x)} \rangle^T \langle \mathbf{s}_i^{(x)} \rangle + trace(\Sigma_{s_i^{(x)}}) \end{aligned} \tag{19}$$

- Update for $\gamma_{\epsilon}^{(x)}$

$$p(\gamma_{\epsilon^{(x)}}|-) \sim \Gamma(\gamma_{s^{(x)}}; e, f) \prod_{i=1}^{N} \mathcal{N}(x_i; \mathbf{D}^{(x)}(z_i \circ s_i^{(x)}), \gamma_{\epsilon^{(x)}}^{-1} I_{P_x}) \tag{20}$$

$\gamma_{\epsilon^{(x)}}$ can be drawn from a Gamma distribution as $p(\gamma_{\epsilon^{(x)}}|-) \sim \Gamma(e', f')$, where

$$\begin{aligned} e' &= e + \frac{1}{2} N \\ f' &= f + \frac{1}{2} \sum_{i=1}^{N} (\|\mathbf{x}_i - \langle \mathbf{D} \rangle (\langle z_i \rangle \circ \langle s_i^{(x)} \rangle)\|^2 + \xi_i) \end{aligned} \tag{21}$$

where

$$\begin{aligned} \xi_i = &\sum_{k=1}^{K} (\langle z_{ik} \rangle \langle s_{ik}^{(x)2} \rangle \langle \mathbf{d}_k^{(x)^T} \mathbf{d}_k^{(x)} \rangle + \langle z_{ik} \rangle^2 \langle s_{ik}^{(x)} \rangle^2 \langle \mathbf{d}_k^{(x)} \rangle^T \langle \mathbf{d}_k^{(x)} \rangle) \\ &+ \sum_{k \neq l} \langle z_{ik} \rangle \langle z_{il} \rangle \Sigma_{s_{i,kl}^{(x)}} \langle \mathbf{d}_k^{(x)} \rangle^T \langle \mathbf{d}_l^{(x)} \rangle \end{aligned} \tag{22}$$

# B Publications

1. **Li He**, Hairong Qi, Russell Zaretzki,"Beta Processing Dictionary Learning for Coupled Feature Space with Application to Image Restoration", *IEEE Transactions on Image Processing*, in submission.

2. **Li He**, Hairong Qi, Russell Zaretzki,"Bayesian Dictionary Learning for Sparse Representation based Image Super Resolution", *Eurasip Journal on Image and Video Processing*, under review.

3. **Li He**, Steve Miller, Hairong Qi, "Improving Neutron Radiograph Image Quality with Advanced Processing Methodologies", *Signal, Image and Video Processing*, under review.

4. **Li He**, Hairong Qi, Russell Zaretzki, "Emotion Transfer for Images Based on Color Combinations", *Signal, Image and Video Processing*, under review [arXiv:1307.3581].

5. Wei Wang, **Li He**, Penn Markham, Hairong Qi, Yilu Liu, "Multiple Event Detection and Recognition through Nonnegative Sparse Unmixing for High-Resolution Situational Awareness in Power Grid", *IEEE Transcations on Smart Grid*, under review.

6. **Li He**, Hairong Qi, Russell Zaretzki,"Beta Processing Dictionary Learning for Coupled Feature Space with Application to Single Image Super Resolution", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2013.

7. Wei Wang, **Li He**, Penn Markham, Hairong Qi, Yilu Liu, "Detection, Recognition, and Localization of Multiple Attacks through Event Unmixing", *IEEE SmartGridComm*, Oct 2013.

8. Wei Wang, Liu Liu, **Li He**, Lingwei Zhan, Hairong Qi, Yilu Liu, "Highly Accurate Frequency Estimation for FNET", *Proceeding of 2013 IEEE Power & Energy Society General Meeting*, Jul 2013.

9. Rui Guo, Shuangjiang Li, **Li He**, Wei Gao, Hairong Qi, Gina Owens, "Pervasive and Unobtrusive Emotion Sensing for Human Mental Health", *7th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, Jun 2013.

10. **Li He**, Steve Miller, Hairong Qi, "Advanced Processing Methodologies Improve Neutron Radiograph Image Quality", *Proceeding of 22nd Annual Research Symposium of American Society for Nondestructive Testing*, Mar 2013

11. **Li He**, Hairong Qi, Russell Zaretzki,"Non-parametric Bayesian Dictionary Learning for Image Super Resolution", *Proceeding of Future of Instrumentation International Workshop*, Nov 2011.

12. Hairong Qi, Yilu Liu, Fangxing Li, Jiajia Luo, **Li He**, Kevin Tomsovic, Leon Tolbert, Qing Cao, "Increasing the Resolution of Wide-Area Situational Awareness of the PowerGrid through Event Unmixing", *Proceeding of Hawaii International Conference on System Sciences*, Jan 2011.

13. **Li He**, Jiajia Luo, Hairong Qi, Chiman Kwan, "A Comparative Study of Several Unsupervised Unmixing algorithms to detecting anomalies in hyperspectral image", *Proceeding of International Symposium on Spectral Sensing Research*, Missouri, Jul 2010.

# Vita

Li He was born in Changsha, Hunan, P. R. China. He started his engineering career in China University of Geoscience, Beijing where he earned a B.E. degree in Electronics and Information Engineering in 2005. Later, he earned a M.S. degree in Electrical Engineering, 2008 from Beijing Jiaotong University. From 2006 to 2008, he worked as an application engineer in Cyan Technology in Cambridge, UK and Hong Kong, China. To pursue a Ph.D., he enrolled the graduate program in the department of Electrical Engineering and Computer Science at the University of Tennessee, Knoxville in fall, 2009. He worked at the Advanced Imaging & Collaborative Information Processing (AICIP) Lab. During his Ph.D. study, his research focused on image processing and machine learning. He completed his Doctor of Philosophy degree in Computer Engineering in Fall 2013 and started working as an algorithm engineer in KLA-Tencor Corporation, California.