UNIVERSITY OF TECHNOLOGY, SYDNEY

DOCTORAL THESIS

Bayesian Nonparametric Modeling and Its Applications

By

Minqi LI

A thesis submitted in fulfilment of the requirements

for the degree of Doctor of Philosophy

in the

School of Computing and Communications

Faculty of Engineering and Information Technology

October 2016

Declaration of Authorship

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signed: _____

Date: _____

Abstract

Bayesian nonparametric methods (or nonparametric Bayesian methods) take the benefit of unlimited parameters and unbounded dimensions to reduce the constraints on the parameter assumption and avoid over-fitting implicitly. They have proven to be extremely useful due to their flexibility and applicability to a wide range of problems. In this thesis, we study the Bayesain nonparametric theory with Lévy process and completely random measures (CRM). Several Bayesian nonparametric techniques are presented for computer vision and pattern recognition problems. In particular, our research and contributions focus on the following problems.

Firstly, we propose a novel example-based face hallucination method, based on a nonparametric Bayesian model with the assumption that all human faces have similar local pixel structures. We use distance dependent Chinese restaurant process (ddCRP) to cluster the low-resolution (LR) face image patches and give a matrix-normal prior for learning the mapping dictionaries from LR to the corresponding high-resolution (HR) patches. The ddCRP is employed to assist in learning the clusters and mapping dictionaries without setting the number of clusters in advance, such that each dictionary can better reflect the details of the image patches. Experimental results show that our method is efficient and can achieve competitive performance for face hallucination problem.

Secondly, we address sparse nonnegative matrix factorization (NMF) problems by using a graph-regularized Beta process (BP) model. BP is a nonparametric method which lets itself naturally model sparse binary matrices with an infinite number of columns. In order to maintain the positivity of the factorized matrices, an exponential prior is proposed. The graph in our model regularizes the similar training samples having similar sparse coefficients. In this way, the structure of the data can be better represented. We demonstrate the effectiveness of our method on different databases.

Thirdly, we consider face recognition problem by a nonparametric Bayesian model combined with Sparse Coding Recognition (SCR) framework. In order to get an appropriate dictionary with sparse coefficients, we use a graph regularized Beta process prior for the dictionary learning. The graph in our model regularizes training samples in a same class to have similar sparse coefficients and share similar dictionary atoms. In this way, the proposed method is more robust to noise and occlusion of the testing images.

The models in this thesis can also find many other applications like super-resolution, image recognition, text analysis, image compressive sensing and so on.

Acknowledgements

Foremost, I wish to appreciate my principle supervisor Professor Xiangjian He for his patience, motivation, constant encouragement and help during my candidature. His guidance helped me in all the time of research. Without his support, I cannot finish my PhD study.

I would also like to express my gratitude to my Co-supervisor Dr. Richard Yida Xu for leading me into the wonderful world of Machine Learning, especially, Bayesian nonparametric models. Without his professional guidance and help, this thesis would not be completed.

I am grateful to Professor Kin-Man Lam in Hong Kong Polytechnic University for his insightful comments and valuable suggestions on my research. I am also grateful to Associate Professor Jian Zhang for his financial support and valuable advice on my PhD study in the first year. I wish to thank my colleagues and the staff in the Faculty of Engineering and Information Technology (FEIT): Qiang Wu, Wenjing Jia, Min Xu, Ruo Du, Sheng Wang, Chao Zen, Muhammad Abul Hasan, Mohammed Ambu Saidi, Khaled W. Aldebei, Ying Wan, Xuhui Fan, Ava Bargi, Shaukat Abidi, etc., for their invaluable help and support. I have the honour of studying and working with them in the past four years which is valuably stamped in my life.

Last but not least, I would like to thank my family for their unconditional support throughout my life.

To my parents for your love and support.

Contents

List of Figures

List of Tables

Abbreviations

GaP	**G**amma **P**rocess
DP	**D**irichlet **P**rocess
CRP	**C**hinese **R**estaurant **P**rocess
ddCRP	**d**istance **d**ependent **C**hinese **R**estaurant **P**rocess
BeP	**B**ernoulli **P**rocess
BP	**B**eta **P**rocess
IBP	**I**ndian **B**uffet **P**rocess
SCR	**S**parse **C**oding **R**ecognition

Symbols

$\mathbb{R}$	the set of reals
$\mathbb{N}$	the set of natural numbers
$\mathbb{E}[\cdot]$	expectation of a random variable
δ_θ	measure concentrated at θ
$\mathbf{c}_i^-$	the assignment set excluding c_i
$DP(\alpha, H)$	Dirichlet process with concentration parameter α and base measure H
$GaP(c, H)$	Gamma process with concentration parameter c and base measure H
$BP(\alpha, H)$	Beta process with positive scalar α and base measure H
$\mathcal{IW}(\cdot)$	normal-inverse-Wishart distribution
$\mathcal{MN}(\cdot)$	matrix-normal distribution
$\mathcal{MT}(\cdot)$	matrix-t distribution
$\|\cdot\|_{lp}$	lp norm
$G_{i,n}$	graph weight between data i and n

Chapter 1

Introduction

1.1 Background

Bayesian modelling methods have been firmly established in machine learning and presented state-of-the-art performance to many applications in computer vision, image processing, data mining and other areas. Most machine learning models are concerned with two problems. One is determining appropriate model classes referred to model selection or model adaptation, such as selecting the number of clusters in a clustering problem, the number of hidden states in a hidden Markov model, the number of latent variables in a latent variable model, or the complexity of features used in nonlinear regression [1]. Another one is learning an appropriate set of parameters within the Bayesian model class from the observed training data. Traditional parametric Bayesian models utilize a finite number of parameters to explain the observations. These models suffer from either over-fitting or under-fitting when the number of parameters or model complexity is not appropriately specified.

Bayesian nonparametric model (or nonparametric Bayesian model), using stochastic processes as prior distributions, is a recently rapidly growing research area in statistics and machine learning. It provides an elegant framework that allows a model to grow with complexity corresponding to the data, with potential infinite parameters [2]. Bayesian nonparametric models constitute an approach to model selection and adaptation, where the sizes of models are allowed to grow with data size. This is opposed to parametric models which uses a fixed number of parameters. More precisely, a nonparametric Bayesian model is a model that [1]:

1. constitutes a Bayesian model on an infinite-dimensional parameter space, and
2. can be evaluated on a finite sample in a manner that uses only a finite subset of the available parameters to explain the sample.

A nonparametric Bayesian model can have infinite-dimensional parameter space. The parameter space is typically chosen as the set of all possible solutions for a given learning problem. In this way, nonparametric Bayesian method overcomes the rigid nature of parametric assumptions and leads to highly flexible inference. On one hand, with the potentially massive parameters, the support of Bayesian nonparametric models is wide enough to avoid under-fitting. On the other hand, proper choice of priors controls the model complexity, hence mitigates over-fitting [3]. These robust properties benefit nonparametric Bayesian models to a wide range of applications including regression, classification, clustering, latent variable modeling, sequential modeling, image segmentation, source separation and so on [1–7].

1.2 Applications and Motivation

In this thesis, we mainly focus on the problems in computer vision and pattern recognition areas by using Bayesian nonparametric models. Some literature reviews of classical Bayesian nonparametric applications are presented as follows.

1.2.1 Image Segmentation

One popular example of Bayesian nonparametric models is Dirichlet process mixture model. It is often applied for clustering, which adapts the number of clusters to the complexity of the data. Its natural extension in image processing and computer vision areas is image segmentation. Typically, the Bayesian nonparametric models for histogram clustering can be used for automatical determining the number of segments. Besides, some spacial constraints can be added to improve the performance. For example, spacial smoothness constraints are considered on the class assignments which are enforced by a *Markov Random Field* [8–10]. Another nonparametric Bayesian model used for image segmentation is distance dependent Chinese restaurant process (ddCRP). The ddCRP clusters data in a biased way: each data point is more likely to

be clustered with other data that are near to it in an external sense. It is particularly well suited for segmenting an image into a collection of contiguous parches [9, 11]. A further extended application of clustering by nonparametric Bayesian models in computer vision is natural scene segmentation. For instance, Pitman-Yor process and hierarchical Pitman-Yor process can be used for natural scene segmentation [12, 13].

As we shown above, Bayesian nonparametric models have been successfully applied on image segmentation problems.

1.2.2 Latent Factor Analysis

Another popular application of Bayesian nonparametric models is latent factor analysis. Beta process factor analysis (BP-FA) model has been well developed for this application. BP-FA allows a dataset to be decomposed into a linear combination of a sparse set of factors, providing information on the underlying structure of the observations [14]. As an important aspect of the BP-FA model is that it is able to enforce sparseness on some subset of the factors, the BP-FA model can be easily extended to the sparse dictionary learning and related applications, such as image interpolation, super-resolution, image denoising, image inpainting and so on [2, 14–17]. For example, Paisley *et al*. presented a Bayesian nonparametric model for image interpolation. The model used the Beta process for dictionary learning, the Dirichlet process for flexibility in dictionary usage and spatial information for encouraging similar dictionary usage within subregions of the image [5, 15].

A similar work presented by Zhou *et al*. considered a nonparametric Bayesian method for image denoising and image interpolation [16]. A truncated Beta-Bernoulli process was employed to infer an appropriate dictionary for the data under test, and also for image recovery. Spatial inter-relationships within imagery were exploited through the use of the Dirichlet and probit stick-breaking processes.

Polatkan *et al*. utilized a Beta-Bernoulli process to learn a set of recurring visual patterns, called dictionary elements from the data for super-resolution problems. A sparse representation of coupled low-resolution image and high-resolution was found by a nonparametric Bayesian method. Then, the coefficients of this representation were used to generate a high-resolution

of the input testing image. The implementations were based on Gibbs sampling and online variational inference [18]. A similar model is presented in [17] for iris image super-resolution.

Related models about latent factor analysis are also presented in other applications. For example, Fox *et al*. used a Beta process prior for discovery sharing features among dynamical systems [6, 19]. Bargi *et al*. proposed a nonparametric conditional factor regression model for domains with multi-dimensional input and response based on Indian buffet process (IBP) enhancements to the latent factors [20]. Knowles and Ghahramani presented an infinite sparse factor analysis and infinite independent components analysis using a distribution over the infinite binary matrix corresponding to the IBP [21].

1.2.3 Other Applications

Besides the above applications, the nonparametric Bayesian methods have also attracted attention to the following problems related to computer vision and pattern recognition.

Fox *et al.* researched Bayesian nonparametric learning of complex dynamical phenomena. A generalization of HDP (hierarchical Dirichlet process)-HMM (hidden Markov model) model was developed, which allowed robust learning of smoothly varying state dynamics through a learned bias towards self-transitions [6]. Similar models are also used for segmentation and classification of sequential data [20].

Sudderth employed nonparametric models for visual object recognition and tracking problems. Dirichlet processes were applied to automatically learn the number of parts underlying each object category, and objects composing each scene [22].

It is well known that matrix completion is closely related to image inpainting, image denoising and recommendation algorithms, like Collaborative Filtering. Paisley *et al.* presented a nonparametric Bayesian model for completing low-rank, positive semi-definite matrices. They proposed a Bayesian hierarchical model to uncover the underlying rank from all positive semi-definite matrices, and complete the matrix by approximating the missing values [23]. Zhou *et al.* proposed to use Beta-Binomial processes for inferring missing values in matrices. The method was based on the low-rank assumption, and the matrix columns were modeled as residing in a non-linear subspace. They provided encouraging performance [24].

The above well applied examples represent the power, flexibility and robustness of Bayesian nonparametric modeling.

1.2.4 Motivations

Although nonparametric Bayesian approaches provide a powerful way of parameter tuning and model selection, and have well developed in many problems as introduced above, there still are some challenging topics in computer vision, image processing and other areas.

In this thesis, we study the foundation theory of Bayesian nonparametric models and extend them to some computer vision and pattern recognition problems for better performance compared with the existing methods. The following section will represent the main work in this thesis.

1.3 Thesis Organization

This thesis presents our work in three areas. We begin by reviewing relevant background and introducing the preliminaries of Bayesian nonparametric models and inference methods in Chapter 2. In Chapter 3, we propose a Face Hallucination method Based on spatial-distance-dependent nonparametric Bayesian learning. Chapter 4 discusses a graph-regularized nonparametric Bayesian approach to sparse nonnegative matrix factorization. In Chapter 5, a graph-regularized nonparametric Bayesian method is presented for face recognition. We summarize our contributions and future work in Chapter 6.

- Preliminaries for nonparametric Bayesian models

  Chapter 2 gives preliminary knowledge about Bayesian nonparametric models and related inference methods. In this chapter, completely random measure and Lévy processes are introduced. Some examples of Lévy processes are presented including the Gamma process, Dirichlet process and Beta process which are related to our later chapters. We introduce the Monte Carlo Markov Chain (MCMC) methods as the inference algorithm of our nonparametric Bayesian methods. They mainly include the Metropolis-Hastings

algorithm and Gibbs sampling algorithm. All these introductions serve the purpose of forming a foundation in understanding the core contributions of this thesis.

- Nonparametric Bayesian model for face hallucination

  In Chapter 3, we propose a novel example-based face hallucination method, based on nonparametric Bayesian learning with the assumption that all human faces have similar local pixel structures. In our method, the low-resolution (LR) face image patches are clustered by using a nonparametric Bayesian method, namely the distance dependent Chinese restaurant process (ddCRP), and the centers of the clusters are calculated. ddCRP does not fix the number of clusters which is different from the traditional methods, but let the image patches themselves to decide. Then, we learn the dictionaries for each cluster to map the LR patches to the high-resolution (HR) patches. The HR patches of the input LR face image can be efficiently generated by using the learned mapping dictionaries. The spatial distance constraint is employed to assist in learning the cluster centers and mapping dictionaries, such that each dictionary can better reflect the detailed information about image patches. Experimental results show that our method is efficient and can achieve competitive performance for face hallucination.

- Nonparametric Bayesian model for sparse nonnegative matrix factorization

  In Chapter 4, we propose a nonparametric Bayesian method for sparse nonnegative matrix factorization (NMF). The traditional NMF algorithms which are mainly based on non-convex optimization are suffering from a drawback that is difficult to maintain its sparse representations. In this chapter, we address the NMF problem by using a graph-regularized nonparametric Bayesian method with a Beta process (BP) prior. Beta process is a nonparametric method which lends itself naturally to model sparse binary matrices with an infinite number of columns. In order to maintain the positivity of the factorized matrices, an element-wise independently and identically distributed exponential prior is proposed. The graph in our model regularizes the similar training samples having similar sparse coefficients and share similar dictionary atoms. In this way, the structure of data can be better represented. An efficient Gibbs sampler is derived to approximate the posterior density of the NMF factors. We demonstrate the effectiveness of our method on different databases. The experimental results show that our proposed method improves the quality of NMF compared with the existing algorithms.

- Nonparametric Bayesian model for face recognition

  In Chapter 5, we propose a graph-regularized nonparametric Bayesian method based on sparse coding recognition (SCR) framework for face recognition (FR) problem. The application of sparse representation for the problem of face recognition has received significant attention. The traditional SCR needs using training samples as the dictionary. However, these dictionaries are not able to accurately reconstruct the testing image by the sparse coefficients when the sparseness in the training data are lacked. In order to get an appropriate dictionary with sparse coefficients for image recognition, we use a graph regularized BP prior for the dictionary learning. The graph in our model regularizes training samples in a same class to have similar sparse coefficients and share similar dictionary atoms, which is beneficial to the reconstruction of the testing image. We demonstrate the effectiveness of our method on different face databases. The experimental results show that our proposed method gives competitive results, especially, on the occluded testing images.

- The final chapter concludes with providing a summary of the contributions in this thesis, and recommendations for future works.

Chapter 2

Preliminaries

Bayesian nonparametric modeling and inference are based on using general stochastic processes as prior distributions. The priors are usually obtained from a class of stochastic processes known as *Lévy processes* and *completely random measures* (CRM). In this chapter, we introduce basic notions on stochastic processes, CRM and Lévy processes, which are preliminaries of nonparametric Bayesian methods. Two remarkable families of Lévy processes: Gamma process and Beta process are studied. Lévy-Khintchine formula and Lévy-Itô decomposition of Lévy process are introduced for giving simple representations of Lévy processes, as well as their relationship with CRM. The last part of this chapter describes an advanced Bayesian inference method, Markov chain Monte Carlo (MCMC), and related sampling algorithms.

2.1 Basic Concepts

Definition 2.1 (Measure space)**.** Let X be a set, Σ be a σ-algebra [1] of its subsets, and $\mu : \Sigma \to [0, +\infty)$ be a measure. A triple (X, Σ, μ) is called a *measure space* if (Σ, μ) is a measurable space and μ is a measure on it.

[1] In mathematical analysis and in probability theory, a σ-algebra on a set X is a collection of subsets of X that includes the empty subset, is closed under complement, and is closed under union or intersection of countably many subsets.

Thus, a measure space implies a measurable space and a measure. The notation (X, Σ, μ) is often shortened to (X, μ) and μ is called a measure on X; sometimes the notation is shortened to X.

Definition 2.2 (Probability space)**.** A *probability space* is a measure space $(\Omega, \mathcal{F}, P)$ in which $P(\Omega) = 1$.

It consists of three parts:

1. A sample space, Ω, which is the set of all possible outcomes;
2. A set of events $\mathcal{F}$, where each event is a set containing zero or more outcomes; and
3. The assignment of probabilities to the events, i.e., a function $P :\rightarrow [0, 1]$ from events to probabilities.

A probability space is a measure space such that the measure of the whole space is equal to one.

Definition 2.3 (Measure)**.** A *measure* on a measurable space $(\Omega, \mathcal{F})$ is a function $\mu : \mathcal{F} \rightarrow \mathbb{R}_+$ such that

1. $\mu(\emptyset) = 0$;
2. If $A_1, A_2, ...$ are disjoint elements of $\mathcal{F}$, then, $\mu(\bigcup_{i=1}^{\infty}(A_i)) = \sum_{i=1}^{\infty} \mu(A_i)$.

Remark. A probability measure P is a measure with $P(\Omega) = 1$.

Definition 2.4 (Random measure)**.** A *random measure* M on $(S, \mathcal{A})$ over the probability space $(\Omega, \mathcal{F}, P)$ is a map $\xi : \mathcal{A} \times \Omega \rightarrow \mathbb{R}_+$, such that

1. For any $A \in \mathcal{A}$, the map $\omega \mapsto M(A, \omega)$ is a random variable; and

2. Almost surely, the map $A \mapsto M(A, \omega)$ is a (probability) measure on $\mathcal{A}$.

Instinctively speaking, random measure are measures drawn from some distributions on measures. That is, random measure M is a measure on a measurable space. At the same time, the random measure M is a random variable on a probability space [25–27]. The expectation of a random measure is called the mean measure, which is denoted as $\nu(A) \triangleq \mathbb{E}[M(A)]$ [28].

A random measure maps from a probability space to a measurable space. It generally might be decomposed as [29]:

$$\mu = \mu_d + \mu_a = \mu_d + \sum_{n=1}^{N} \kappa_n \delta_{X_n} \tag{2.1}$$

Here, μ_d is a diffuse measure (non-atomic measure) without atoms, while μ_a is a purely atomic measure [2].

Random Measure Construction

The following theorem presents an example of random measure construction.

Theorem 2.1.1. *If $\pi_1, ..., \pi_k$ are i.i.d Beta$(\alpha/k, 1)$ random variables, their order statistics $\pi_{(1)} \geq \pi_{(2)} \geq ... \geq \pi_{(k)}$ satisfy*

$$\pi_i = \prod_{j=1}^{i} \beta_j, \tag{2.2}$$

where $\beta_1, \beta_2, ..., \beta_k$ are independent and $\beta_i \sim$ Beta$(\alpha(k-i+1)/k, 1)$ random variables. Then, π_i are random measures.

Stick-breaking is a popular method for constructing nonparametric models. As a Beta$(\alpha, 1)$ fraction can be broken from a unit stick, from Theorem 2.1.1, a stick-breaking method for constructing random measure can be represented as:

$$\begin{aligned} &\beta_k \sim Beta(1, \alpha_0), \quad k = 1, 2, ..., \\ &\pi_1 = \beta_1, \quad \pi_2 = \beta_2, \quad ..., \quad \pi_k = \beta_k \prod_{l=1}^{k-1} (1 - \beta_l), \\ &G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}, \end{aligned} \tag{2.3}$$

where ϕ_k are independently drawn from a base distribution G_0 in some space. Then, G is a random measure (actually a probability measure) as shown in Fig. 2.1.

[2] In mathematics, more precisely in measure theory, an atom is a measurable set which has positive measure and contains no set of smaller but positive measure.

FIGURE 2.1: Stick breaking for constructing a random measure

Example 2.1.2 (Random counting measure)**.** *A random measure of the form* $\mu = \sum_{n=1}^{N} \delta_{X_n}$*, where* $N \in \mathbb{N}$*, and* δ *is the Dirac measure* [3]*, and* X_n *are random variables, is called a* point process *or* random counting measure.

This is a simple natural random measure. It describes the set of N particles, whose locations are given by the random variables X_n. The diffuse component μ_d is null for a counting measure [29]. In a formal notation of the above, a random counting measure is a map from a probability space $(\Omega, \mathcal{F}, P)$ to the measurable space $(S, \mathcal{A})$. Here, S is the set of locally finite counting measures with finite integer-valued measures, and $\mathcal{A}$ is σ-algebra of its Borel sets [30].

Poisson Random Measure

Example 2.1.3 (Poisson random measure)**.** *Let* $(S, \mathcal{A})$ *be a measurable space and let* v *be a measure on it. A random measure* M *on* $(S, \mathcal{A})$ *is said to be Poisson with mean* v *if*

1. *for every* A *in* $\mathcal{A}$*, the random variable* $M(A)$ *has the Poisson distribution with mean* $v(A)$*; and*
2. *whenever* $A_1, ..., A_n$ *are in* $\mathcal{S}$ *and disjoint, the random variables* $M(A_1), ..., M(A_n)$ *are independent for every* $n \geq 2$*.*

In particular, the Poisson random measure is a positive integer-valued random measure. It can be constructed as the counting measure of randomly scattered points, as shown by the following proposition [31, 32].

[3] Dirac measure:

$$Dirac : \delta_x(A) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$$

i.e. a point measure (or delta measure) at point x.

Proposition 2.1.4. *Suppose that v is a measure on (E, ξ) such that $v(E) < \infty$. Then, there exists a Poisson random measure with mean measure v.*

The proof of the this proposition is referred to [32, 33]. Actually, Poisson random measure is a foundation of other Lévy processes and nonparametric Bayesian models. It is often used for describing jumps of stochastic processes, in particular in Lévy-Itô decomposition of the Lévy processes which will be introduced in the later sections.

2.2 Completely Random Measure

Definition 2.5 (Completely random measure)**.** Let H be a random measure on $(\Omega, \mathcal{F})$, we say that H is *completely random* if for any set of disjoint measurable sets $F_1, ..., F_m$ of $\mathcal{F}$, the random variables $H(F_1), ..., H(F_m)$ are mutually independent.

Completely random measures (CRM) were introduced by Kingman in 1963 as a generalization of a stochastic process with independent increments to arbitrary index sets. Kingman showed that any completely random measure can be represented as the sum of three components based on the nonhomogeneous Poisson process [27]:

1. A non-random measure (deterministic component);
2. An atomic measure with (at most countable) fixed atom locations and random atom masses; and
3. An atomic measure with random atom locations and random atom masses (ordinary component).

The ordinary component sometimes is described as a discrete measure $\sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}$, where the set of points $(\pi_k, \theta_k)_{k \geq 1}$ is distributed as an inhomogeneous Poisson point process on $\mathbb{R}_+ \times \Omega$. More formally, we have the following theorem.

Theorem 2.2.1 (CRM)**.** *A CRM μ can be decomposed into a sum of three independent components:*

$$\mu = \mu_0 + \sum_{j=1}^{J} v_j \delta_{x_j} + \sum_{k=1}^{K} \pi_k \delta_{\theta_k}, \tag{2.4}$$

where μ_0 is a non-random measure, J is constant, $\{x_j|j = 1, 2, ..., n\}$ are fixed members of Ω, $v_j|j = 1, 2, ..., n$ are mutually independent random variables on $\mathbb{R}_+$ and $N = \sum_{k=1}^{K} \pi_k \delta_{\theta_k}$ is a Poisson process over $\mathbb{R}_+ \times \Omega$.

This construction has significant consequences for Bayesian modeling and computation. In particular, it allows connections to be made to the exponential family and to conjugacy [34]. CRM has become a key concept in Bayesian nonparametric statistics. Many nonparametric priors are random probability measures or random measures. Most of these random measures are either completely random measures, or are obtained from a completely random measure by normalization (e.g. Dirichlet process) [35].

2.3 Lévy Process

In probability theory, a Lévy process, named after the French mathematician Paul Lévy, is a stochastic process with independent, stationary increments. Before giving a formal definition of Lévy process, we first introduce stochastic process with some structures.

Definition 2.6 (Stochastic process)**.** Given a probability space $(\Omega, \mathcal{F}, P)$ and a measurable space $(S, \mathcal{S})$, any collection of random variables $X = \{X_t, t \in T\}$ defined on $(\Omega, \mathcal{F}, P)$ is called a stochastic process with index set T.

In order to make mathematical models less complicated, some simplifying assumptions or some simplifying structures (restrictions) can be added. There are some popular dependence structures put on stochastic processes which mathematicians have developed and used for years. Here, we introduce the concept of *filtration*, a structure on a probability measure, of which concept is often used in Lévy process [36].

Definition 2.7 (Filtration)**.** Let $(\Omega, \mathcal{F}, P)$ be a probability space. A *filtration* on $(\sigma, \mathcal{F}, P)$ is an increasing family $(\mathcal{F})_{t\geq 0}$ of sub-σ-algebras of $\mathcal{F}$. In other words, for each t, $\mathcal{F}_t$ is a σ-algebra included in $\mathcal{F}_t$ and if $s \leq t$, $\mathcal{F}_s \subset \mathcal{F}_t$. A probability space $(\Omega, \mathcal{F}, P)$ endowed with filtration $(\mathcal{F})_{t\geq 0}$ is called a filtered probability space.

Intuitively speaking, a filtration is an increasing information flow about $(X_{t\in[0,T]})$ as time progresses [37].

Definition 2.8. [Lévy process] A Càdlàg [4], real valued stochastic process $L = (L_t)_{0 \leq t \leq T}$ with $L_0 = 0$ a.s. is called a *Lévy process* if the following conditions are satisfied:

1. L has independent increments, i.e. $L_t - L_s$ is independent of $\mathcal{F}_s$ for any $0 \leq s < t \leq T$.
2. L has increments, i.e. for any $0 \leq s, t, \leq T$ the distribution of $L_{t+s} - L_t$ does not depend on t.
3. L is stochastically continuous, i.e. for every $0 \leq s, t, \leq T$ and $\epsilon > 0$: $lim_{s \to t} P(|L_t - L_s| > \epsilon) = 0$.

Note. From the definition of Lévy process, we can see that Lévy processes have Càdlàg property. This property is important for constructions of Lévy processes which will be presented in the later section of this chapter. The properties of stationary and independent increments implies that a Lévy process is a Markov process [38, 39].

Lévy process represents the motion of a point whose successive displacements are random and independent, and statistically identical over different time intervals of the same length. The simplest Lévy process is the linear drift, a deterministic process. Brownian motion is the only (non-deterministic) Lévy process with continuous sample paths. Some special cases of Lévy processes in discrete time are random walks [5] [38].

2.3.1 Lévy Process and Infinite Divisible Distribution

From Definition 2.8, it is difficult to see how rich a class of Lévy processes can form. The following notion of an *infinitely divisible distribution* will show that they have a close relationship with Lévy processes [36].

Definition 2.9 (Infinite divisible distribution)**.** A real valued random variable X with the probability density function $P(x)$ is said to be infinitely divisible if for $\forall n \in N$ there exist i.i.d random variables $X_1, X_2, ..., X_n$ satisfying:

$$X = X_1 + X_2 + ... + X_n. \tag{2.5}$$

[4] i.e. right continuous with left limits.

[5] A discrete-time process $(X_n)_{n=0,1,2,...}$ with stationary and independent increments is a Random Walk: $X_n = X_0 + \sum_{j=1}^{n} \eta_j$ with i.i.d. increments $\eta_j = X_j - X_{j-1}$.

$P(x)$ is said to be an infinitely divisible distribution.

Example 2.3.1 (Normal distribution)**.** *A random variable $X \sim \mathcal{N}(\mu, \sigma^2)$ is infinitely divisible, with the distributions*

$$X^{(n)} \sim \mathcal{N}(\mu/n, \sigma^2/n). \tag{2.6}$$

Proof. As the characteristic function [6] of X with distribution $\mathcal{N}(\mu/n, \sigma^2/n)$ is

$$\begin{aligned}
\psi(u) &= \mathbb{E}[e^{iuX}] \\
&= \int_{\mathbb{R}} e^{iux} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x-\mu)^2}{2\sigma^2}) \\
&= e^{iu\mu - \frac{u^2\sigma^2}{2}} \int_{\mathbb{R}} e^{iux} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x-(\mu+i\sigma^2 u))^2}{2\sigma^2}) \\
&= e^{iu\mu - \frac{u^2\sigma^2}{2}},
\end{aligned} \tag{2.7}$$

it can be seen that, the characteristic function of X satisfying $\exp(iu\mu - u^2\sigma^2) = \exp(iu\mu/n - u^2(\sigma/\sqrt{n})^2/2)^n)$, which is a characteristic function of the summary of n i.i.d normal variables with a mean of μ/n and standard deviation of $\sigma/\sqrt{n}$. □

Example 2.3.2 (Poisson distribution)**.** *A Poisson random variable X with parameter λ is infinitely divisible.*

Proof. As the characteristic function of a Poisson random variable X with parameter λ is of the form

$$\psi(u) = \sum_{k=0}^{\infty} \frac{e^{-\lambda}(\lambda e^{iu})^k}{k!} = e^{\lambda(e^{iu}-1)}, \tag{2.8}$$

it can be seen that $e^{\lambda(e^{iu}-1)} = e^{(\lambda(e^{iu}-1)/n)^n}$. □

The above two simple examples of infinitely divisible distributions are Normal and Poisson distributions, which are often used for constructing Lévy processes.

Lemma 2.10. *A Lévy Process L_t is infinitely divisible for each $t \geq 0$.*

[6] refer Definition 2.11

Proof. Let L_t be a Lévy Process, then for each $n \in \mathbb{N}$, $L_t = L_{t/n} + (L_{\frac{2t}{n}} - L_{\frac{t}{n}}) + ... + (L_{\frac{nt}{n}} - L_{\frac{(n-1)t}{n}})$. □

As Lemma 2.10 represents, there is a deep connection between Lévy process and infinitely divisible random variables. The following proposition will give a more strong conclusion.

Proposition 2.3.3. *If $X_{t\in[0,\infty)}$ is a real valued Lévy process on a filtered probability space $(\Omega, \mathcal{F}_{t\in[0,T]}, P)$, then X_t has a infinitely divisible distribution for $\forall t \in [0,T]$. Inversely, for every infinitely divisible distribution P on $\mathbb{R}$, there exists a Lévy process $(X_{t\in[0,\infty)})$ on $\mathbb{R}$ whose distribution of increments $X_{t+h} - X_t$ is governed by P.*

This corresponds to Corollary 11.6 of [40]. This proposition presents that there is one-to-one correspondence between an infinitely divisible and a Lévy process. We can get that every Lévy process can be associated with the law of an infinitely divisible distribution. Inversely, given any random variable X, whose law is infinitely divisible, we can construct a Lévy process. Table 3.2 represents the one-to-one correspondence between an infinitely divisible distribution and a Lévy process [36].

Normal distribution	Brownian motion (with drift)
Poisson distribution	Poisson process
Compound Poisson distribution	Compound Poisson process
Exponential distribution	Gamma process
Cauchy distribution	Cauchy process

TABLE 2.1: Relationship between Lévy process and infinitely divisible distribution.

2.3.2 Lévy-Khintchine Representation

In probability theory and statistics, the probability density function of any real-valued random variable can be completely defined by the characteristic function, which is the inverse Fourier transform of the probability density function. There are particularly simple results for the characteristic functions of the probability distributions. Here, we discuss the characteristic exponents of Lévy processes.

Definition 2.11 (Characteristic function)**.** A *characteristic function* ψ of X is defined by

$$\psi(u) = \int_{\mathbb{R}} e^{iux} P(X \in dx), \tag{2.9}$$

where the function log ψ is referred to the characteristic exponent of X.

A simple representation of a Lévy process is given by Lévy-Khintchine representation.

Theorem 2.3.4 (Lévy-Khintchine formula for Lévy processes)**.** *For every Lévy process* $L = (L_t)_{0\leq t\leq T}$*, we have that* $\mathbb{E}[e^{iuL_t}] = e^{t\Psi(u)} = exp[t(ibu - \frac{u^2c}{2} + \int(e^{iux} - 1 - iux\mathbf{l}_D)v(dx))]$ *where* $\Psi(u)$ *is the characteristic exponent of* L_t*, a random variable with an infinitely divisible distribution.*

$\Psi(u)$ is called a characteristic exponent (or a log characteristic function) given by:

$$\Psi(u) = iub - \frac{u^2c}{2} + \int (e^{iux} - 1 - iux\mathbf{l}_D)v(dx), \tag{2.10}$$

where $D = \{x : |x| \leq 1\}$, c is a unique nonnegative constant (i.e. $c \in \mathbb{R}_+$), b is a unique constant on $\mathbb{R}$, and v is a unique measure on $\mathbb{R}$ satisfying the following two conditions [36]:

1. $v(\{0\}) = 0$; and
2. $\int_{-\infty}^{+\infty} \min\{|x|^2, 1\}v(dx) < \infty$ (mathematicians write this as $\int_{-\infty}^{+\infty}(1 \wedge |x|^2)vd(x) < \infty$).

Here, $\int_{-\infty}^{+\infty}(1 \wedge |x|^2)vd(x) < \infty$ means two things: $v(|x| > \epsilon) < \infty$ for any $\epsilon > 0$, and $\int_{(-1,1)} x^2 v(dx) < \infty$.

The proof of this theorem is complicated. Interested readers are referred to [40].

Note. These two conditions ensure that the integral in the Lévy-Khintchine representation converges since the integrand is $O(1)$ for $|x| \geq 1$ and $O(x^2)$ for $|x| < 1$. When the jump size is greater than 1 (i.e. $|x| > 1$), under the second condition, the jump size is considered to be a large jump using the arbitrary truncation point of 1. Then, Condition 2 becomes $\int_{|x|>1} v(dx) < \infty$, which means that the expected number of large jumps per unit of time is finite. When the jump size is less than 1 (i.e. $|x| < 1$), the jump is considered to be a small jump. Then, Condition 2 becomes $\int_{|x|<1} |x|^2 v(dx) < \infty$, which means that the measure must be square-integrable around the origin [36]. Otherwise, the corresponding Lévy process should have infinite many jumps of size greater than ε in finite time. However, this contradicts with the Càdlàg (right continuous with left limits) property of the paths in definition 2.8. Therefore, the square integrability condition controls the intensity of small jumps. It is crucial for the construction of a Lévy process with jump intensity ν [41].

This is the general Lévy-Khintchine representation of all Lévy processes. Sometimes, the Lévy processes are restricted by additional condition to the measure v, which is $\int(1\wedge|x|)\nu d(x)<\infty$ [39, 42].

2.3.3 Lévy-Itô Decomposition of Lévy processes

From Theorem 2.3.4, we can see that any Lévy process on $\mathbb{R}^d$ can be characterized by three quantities: a non-negative definite symmetric matrix a, a vector b, and a σ-finite measure ν. The Lévy-Itô decomposition gives an explicit representation of a Lévy process with characteristics (a,b,ν). Firstly, we introduce Lévy triplet [36].

Definition 2.12 (Lévy triplet (generating triplet))**.** In the Lévy-Khintchine formula, a unique nonnegative constant b is called a Gaussian variance and a unique real valued measure v satisfying $v(\{0\})=0$ and $\int_{\mathbb{R}}(1\wedge|x|^2)vd(x)<\infty$ is called a Lévy measure. A unique real valued constant c does not have any intrinsic meaning since it depends on the behavior of a Lévy measure v. It turns out that these triplets uniquely define a Lévy process as a result of Lévy-Itô decomposition. This triplet is called a Lévy-Khintchine triplet and compactly written as (b,c,v).

Theorem 2.3.5 (Lévy-Itô Decomposition [38])**.** *Consider a triplet (b,c,v) where $b\in\mathbb{R}$, $c\in\mathbb{R}_{\geq 0}$ and v is a measure satisfying $v(\{0\})$=0 and $\int_{\mathbb{R}}(1\wedge|x|^2)vd(x)<\infty$. Then, there exists a probability space $(\Omega,\mathcal{F},P)$ on which four independent Lévy processes exist, $L^{(1)}$,$L^{(2)}$,$L^{(3)}$,$L^{(4)}$, where $L^{(1)}$ is a constant drift, $L^{(2)}$ is a Brownian motion, $L^{(3)}$ is a compound Poisson process and $L^{(4)}$ is a square integrable (pure jump) martingale [7] with an a.s. countable number of jumps of magnitude less than 1 on each finite time interval. Taking $L=L^{(1)}+L^{(2)}+L^{(3)}+L^{(4)}$, we have that there exists a probability space on which a Lévy process $L=(L_t)_{0\leq t\leq T}$ with characteristic exponent $\Psi(u)=iub-\frac{1}{2}u^2c+\int_{-\infty}^{+\infty}(e^{iux}-1-iux\mathbf{I}_{|x|<1})v(dx)$ for all $u\in\mathbb{R}$, is defined.*

[7] Martingale: Consider a filtered probability space $(\Omega,\mathcal{F},P)$. A Càdlàg stochastic process $(X_t)_{t\in[0,T]}$ is said to be a martingale with respect to filtration $\mathcal{F}_t$ and under the probability measure P if it satisfies the following conditions [36]:

1. X_t is nonanticipating.
2. $\mathbb{E}[|X_t|]<\infty$ for $\forall t\in[0,T]$. Finite mean condition.
3. $\mathbb{E}[X_u|\mathcal{F}_t]=X_t$ for $\forall u>t$.

The Lévy-Itô decomposition is a hard mathematical result to prove. Interested readers are referred to Chapter 4 in [40] or Chapter 2 in [38].

Lévy-Itô decomposition basically states that every sample path of a Lévy process can be represented as a sum of two independent processes: one is a continuous Lévy process and the other is a compensated sum of independent jumps. Obviously, a continuous Lévy process is a Brownian motion with drift (this is the only continuous Lévy process) [36].

By taking linear combinations of Lévy jump processes and Gaussain Lévy processes, we can obtain all Lévy processes. This is the content of the Lévy-Itô decomposition theorem, which can help us construct different Lévy processes.

2.3.4 Lévy Measure

The structure of jumps of a Lévy process can be determined by its Lévy measure. In this section, we will represent some examples of Lévy processes and discuss their Lévy measures.

Definition 2.13 (Lévy measure of all Lévy processes)**.** Let $(X_{t\in[0,\infty)})$ be a real valued Lévy process with Lévy triplet (b, c, v) defined on a filtered probability space $(\Omega, \mathcal{F}, P)$. The Lévy measure v of a Lévy process $(X_{t\in[0,\infty)})$ is defined as a unique positive measure on $\mathbb{R}$ which measures (counts) the expected (average) number of jumps per unit of time:

$$v(\Omega) = E[\sharp\{t \in [0,1] : \Delta X_t = X_t - X_{t-} \neq 0, \Delta X_t \in \Omega\}], \tag{2.11}$$

where $\Delta X_t \in \Omega$ indicates that the jump size belongs to a set Ω and Ω is a member of Borel set [8] [36].

In brief, the Lévy measure v is a measure on $\mathbb{R}$ that satisfies $v(\{0\}) = 0$ and $\int_{(-1,1)} x^2 v(dx) < \infty$. Intuitively speaking, it describes the expected number of jumps of a certain height in a time interval of length 1 [38].

From the definition, we can see that, the Lévy measure is more flexible than probability measure who constrains the measure satisfying $\int_{\mathbb{R}} v\mathrm{d}(x) = 1$. Therefore, if v is a finite measure, i.e. $\lambda := v(\mathbb{R}) = \int_{\mathbb{R}} v\mathrm{d}(x) < \infty$, we can normalize the the measure by $F(\mathrm{d}x) := \frac{v\mathrm{d}(x)}{\lambda}$, which is a

[8] Given a topological space, the Borel set an element of the σ-algebra generated by the open sets.

probability measure. Here, λ is the expected number of jumps and $F(\mathrm{d}x)$ is the distribution of the jump size. If $v(\mathbb{R}) = \infty$, then an infinite number of (small) jumps is expected [38].

2.3.5 Examples of Lévy Processes

To conclude our introduction of Lévy processes and infinite divisible distributions, the following section will represent some concrete examples of Lévy processes.

Brownian Motion

Example 2.3.6 (Brownian motion)**.** *A standard Brownian motion $B_{t\in[0,\infty)}$ defined on a filtered probability space $(\Omega, \mathcal{F}, P)$ is a Lévy process satisfying*

1. *its increments are independent and stationary;*
2. *for $t \geq 0$, B_t are continuous functions of t;*
3. *the process starts from 0 almost surely, i.e. $P(B_0 = 0) = 1$;*
4. *$B_t \sim N(0, t)$, its increments follow a Gaussian distribution with the mean 0 and the variance t.*

A Brownian motion with drift satisfies the conditions of the Lévy process. Actually, it is an only Lévy process with continuous sample paths [36]. Fig. 2.2 shows a standard Brownian motion with drift 0.

The simplest Lévy process is the linear drift, a deterministic process. Brownian motion is the only (non-deterministic) Lévy process with continuous sample paths. A more general Brownian motion with drift can be defined as

$$(X_{t\in[0,\infty)}) \equiv (\mu t + \sigma B_{t\in[0,\infty)}), \tag{2.12}$$

where $B_{t\in[0,\infty)}$ is a standard Brownian motion.

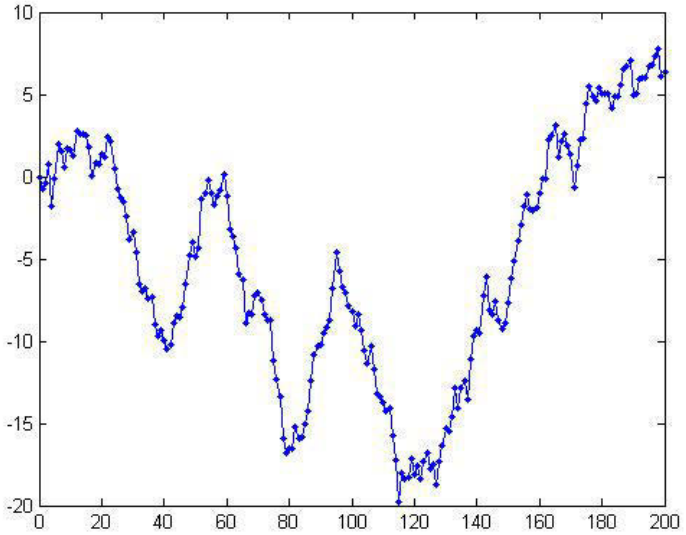


FIGURE 2.2: An example of Lévy processes: standard Brownian motion

Lévy Measure of Brownian Motion

Proposition 2.3.7. *Suppose* $(X_{t\in[0,\infty)}) \equiv (\mu t + \sigma B_{t\in[0,\infty)})$ *is a Brownian motion with drift, where* $B_{t\in[0,\infty)}$ *is a standard Brownian motion,* $X_t \sim \mathcal{N}(\mu t, \sigma^2 t)$*, then, the Lévy measure of* $X_{t\in[0,\infty)}$ *is 0.*

Proof. The characteristic function of $X_{t\in[0,\infty)}$ is

$$\begin{aligned}\psi_{X_t}(u) &= \int_{-\infty}^{\infty} e^{iux} \frac{1}{\sqrt{(2\pi\sigma^2 t)}} \exp(-\frac{(x-ut)^2}{2\sigma^2 t}) dx \\ &= \exp(0 \cdot t + i\mu t u - \frac{\sigma^2 t u^2}{2}).\end{aligned} \tag{2.13}$$

It can be seen that the Lévy triplet of $X_{t\in[0,\infty)}$ is $(b = \mu, c = \sigma^2, v = 0)$, and the Lévy measure is 0. □

Poisson Process

The most elementary example of a pure jump Lévy process in continuous time is the Poisson process. It takes values in $\{0, 1, 2, ...\}$ and jumps up one unit each time after an exponentially distributed waiting time.

Definition 2.14 (Poisson process)**.** Let $\lambda > 0$. The counting process $N(t), \{t \in [0, \infty)\}$, is called a *Poisson process* with rates (or intensity) λ, such that,

1. $N(0) = 0$;
2. $N(t)$ has independent increments; and
3. the number of arrivals having any interval of length $\tau > 0$ has Poisson$(\lambda\tau)$ distribution.

Here, counting process is a random process $\{N(t), t \in [0, \infty)\}$ if $N(t)$ is the number of events occurred from time 0 up to and including time t. For a counting process, we assume that $N(0) = 0$, $N(t) \in \{0, 1, 2, ...\}$, for all $t \in [0, \infty)$. For $0 \leq s \leq t$, $N(t) - N(s)$ shows the number of events that occur in the interval $(s, t]$.

Note. A Poisson process, $\{N_t : n \geq 0\}$ with intensity λ, is a Lévy process with distribution at time $t > 0$, which is Poisson distribution with parameter λt. An easy calculation reveals that

$$\mathbb{E}(e^{i\theta N_t}) = e^{-\lambda t(1-e^{i\theta})}, \tag{2.14}$$

and hence, its characteristic exponent is given by $\psi(\theta) = \lambda(1 - e^{i\theta})$ for $\theta \in \mathbb{R}$.

Compound Poisson Process

A more general discontinuous Lévy process is the compound Poisson process.

Definition 2.15 (Compound poisson process)**.** $\{Y(t), t \geq 0\}$ is a *compound Poisson process* if $Y(t) = \sum_{i=1}^{N(t)} D_i$ where $D_1, D_2, ...$ are independent and identically distributed random variables with distribution f_D, and these random variables are also independent from $\{N(t), t \geq 0\}$, a Poisson process.

Compound Poisson processes are pure jump Lévy processes, i.e., the paths are constant apart from a finite number of jumps in finite time. The Poisson process is a special case of the compound Poisson process where D_i equals to 1.

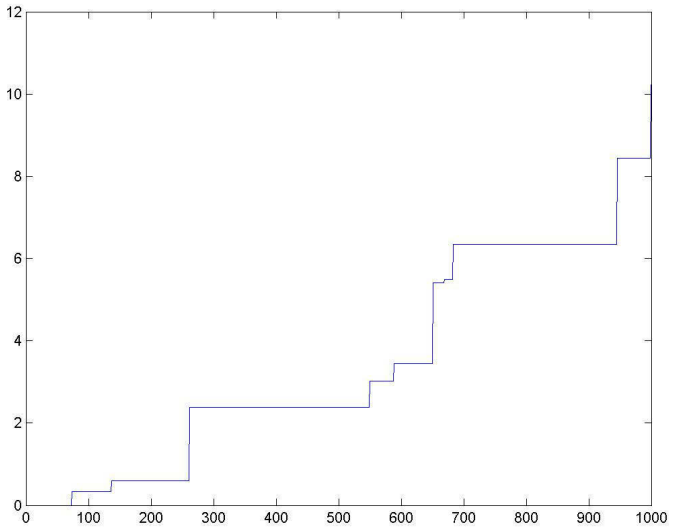


FIGURE 2.3: A sample of compound Poisson process.

Fig. 2.3 shows an example of compound Poisson process with $\lambda = 1$.

Lévy Measure of Compound Poisson Process

Lemma 2.3.8. *Let N be a Poisson distributed random variable with parameter $\lambda \geq 0$ and $J = (J_k)_{k\geq 1}$ be an i.i.d sequence of random variables with law (probability measure) f. Then, the characteristic function of a compound Poisson distributed random variable is*

$$\mathbb{E}[e^{iu\sum_k=1^N J_k}] = exp(\lambda \int_{\mathbb{R}} (e^{iux} - 1) f(dx)). \tag{2.15}$$

Proposition 2.3.9 (Lévy measure of a compound Poisson process)**.** *Suppose that $N = \{N_t : t \geq 0\}$ is a Poisson process with intensity λ and consider a compound Poisson process $\{X_t : t \geq 0\}$ defined by $X_t = \sum_{i=1}^{N_t} \xi_i, t \geq 0$. Using the fact that N has stationary independent increments together with the mutual independence of the random variables $\{\xi_i : i \geq 1\}$, for $0 \leq s < t < \infty$, by writing*

$$X_t = X_s + \sum_{i=N_s+1}^{N_t} \xi_i$$

based on Lemma 2.3.8, we have the Lévy-Khinchine formula for a compound Poisson process taking the form

$$\Psi(u) = \lambda \int_{\mathbb{R}} (1 - e^{iux}) f(dx). \tag{2.16}$$

The Lévy triplet of $X_{t\in[0,\infty)}$ *is* $(b = 0, c = 0, v = \lambda f(x))$. *The Lévy measure is given by* $v(x) = \lambda f(x)$.

Note. The Lévy measure of a compound Poisson process is always finite with total mass equal to the rate λ of the underlying process N [43].

Poisson process and compound Poisson process are basic processes of Lévy processes with infinite jumps. The jump part of a Lévy process can be recovered from these counting measure valued processes by integration, i.e. summation of the jump sizes. Many Lévy processes with infinite jump intensity can be constructed by related Poisson processes.

A family of Lévy processes, the pure-jump nondecreasing Lévy processes fit into the category of the completely random measure. The Beta process can be regarded as an example of such a process. The Dirichlet process is a normalization of Gamma process who falls into this family as well [36, 44]. In this thesis, we focus on some popular Lévy processes for machine learning and computer vision applications, especially, the applications related to Gamma process, Dirichlet process and Beta process.

Gamma Process

A Gamma process is a random process with independent Gamma distributed increments. It is a pure-jump increasing Lévy process. An homogeneous Gamma process with shape parameter α and scale parameter β is a stochastic process $X(t), t \geq 0$ on $\mathbb{R}_+$ such that [45]:

1. X(0)=0,
2. X(t), $t \geq 0$ is s stochastic process with independent increments, and
3. for $0 \leq s < t$, the distribution of the random variable $X(t) - X(s)$ is the Gamma distribution $\Gamma(\alpha(t-s), \beta)$.

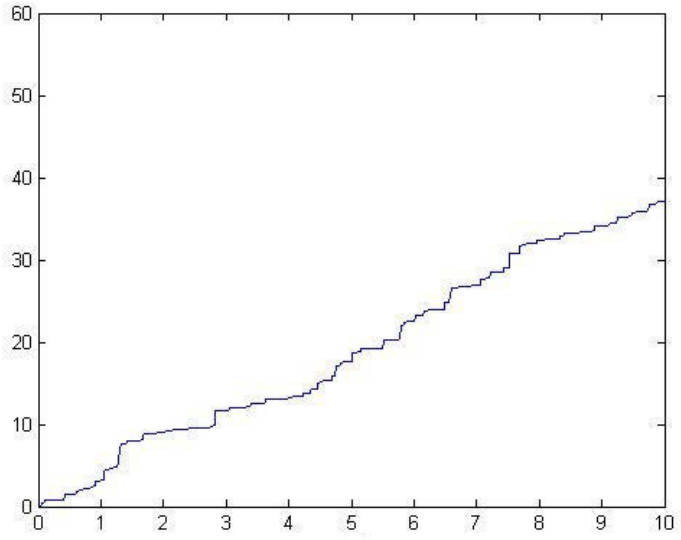


FIGURE 2.4: A sample of Gamma process with $\alpha = 4$, $\beta = 1$.

Fig. 2.4 is a Gamma process with parameters $\alpha = 4$ and $\beta = 1$.

The Gamma process is a pure-jump increasing Lévy process. The jumps size lying in the interval $[x, x + dx]$ occurs as a Poisson process with intensity $\nu(x)dx$. The parameter β controls the rate of jump arrivals and the scaling parameter α inversely controls the jump size [46].

The following definition describes Gamma process in a completely random measure view.

Definition 2.16 (Gamma process)**.** $G \sim \text{GaP}(\alpha, H)$ is a completely random measure defined on the product space $\mathbb{R}_+ \times \Omega$, with concentration parameter α and a finite and continuous base measure H over a complete separable metric space Ω. If $G(A_i) \sim \text{Gamma}(H(A_i), 1/\alpha)$, $i \in \mathbb{N}$ are independent Gamma random variables for disjoint partition $\{A_i\}$ of Ω, then $G(A)$ is a *Gamma process*.

Lévy Measure of Gamma Process

To get the Lévy measure of a Gamma process, we introduce the following Lemmas [47].

Lemma 2.3.10 (Frullani's Integral)**.** *if $f'(x)$ is continuous and the integral converges,*

$$\int_0^\infty \frac{f(ax) - f(bx)}{x} dx = [f(0) - f(\infty)] ln(\frac{b}{a}), \tag{2.17}$$

where $a, b > 0$.

Lemma 2.3.11. *For all $\alpha, \beta > 0$ and $z \in C$, we have*

$$(1 - \frac{iu}{\alpha})^{-\beta} = e^{-\int_0^\infty (1-e^{ux})\beta x^{-1} e^{-\alpha x} dx}. \tag{2.18}$$

Proof. Let $f(x) = e^{-x}$, $\alpha = a > 0$, and $b = \alpha - u$. From Lemma 2.3.10, we have

$$(1 - \frac{iu}{\alpha})^{-\beta} = e^{-\int_0^\infty (1-e^{ux})\beta x^{-1} e^{-\alpha x} dx}.$$

□

Proposition 2.3.12 (Lévy measure of Gamma process)**.** *Let $G \sim GaP(\alpha, G_0)$ be a Gamma process on the product space $\mathbb{R}_+ \times \Omega$, with scale parameter $1/\alpha$ and base measure G_0. The Lévy measure of GaP(α, G_0) is $v(dxd\omega) = x^{-1}e^{-\alpha x}dxG_0(d\omega)$.*

Proof. For $\alpha, \beta > 0$, the probability density function of Gamma distribution is:

$$f(x, \alpha, \beta) = \frac{\alpha^\beta}{\Gamma(\beta)} x^{\beta-1} e^{-\alpha x}, x \in (0, +\infty)$$

concentrated on $(0, \infty)$. The characteristic function of $f(x, \alpha, \beta)$ is

$$\psi(u) = \int_0^{+\infty} e^{iux} f(x, \alpha, \beta) dx = \int_0^{+\infty} \frac{\alpha^\beta x^{\beta-1}}{\Gamma(\beta)} e^{-x(\alpha-iu)} dx. \tag{2.19}$$

Let $y = x(\alpha - iu)$, we have

$$\psi(u) = \frac{\alpha^\beta}{\Gamma(\beta)(\alpha - iu)^\beta} \int_0^{+\infty} y^{\beta-1} e^{-y} dy = (1 - \frac{iu}{\alpha})^{-\beta}. \tag{2.20}$$

From Formula 2.21 and Lemma 2.3.11, we can get that the Lévy-Khintchine expression of a Gamma process has the characteristic exponent with

$$\Psi(u) = \beta \int_0^\infty (1 - e^{iux}) x^{-1} e^{-\alpha x} dx. \tag{2.21}$$

The Lévy measure of a Gamma process GaP(x, α, G_0) defined on the product space $\mathbb{R}_+ \times \Omega$ is $\nu(dxd\omega) = x^{-1}e^{-\alpha x}dxG_0(d\omega)$. □

Gamma process is a fundamental process in Bayesian nonparametric methods. It has almost surely a finite and positive measure, so it is possible to normalize the measure to obtain a probability measure. This allows us to define a normalized completely random measure $\widetilde{G} := G/G(\Omega)$. Actually, the normalized Gamma process is a Dirichlet process (DP), which will be presented next.

Dirichlet Process

The most popular nonparametric Bayesian method is Dirichlet process (DP). It is considered as a distribution over distributions, i.e., each sample from Dirichlet process is a distribution, not just a variable.

Definition 2.17 (Dirichlet process)**.** Let H be a distribution over Θ and α be a positive real number. Then, for any finite measurable partition $A_1, A_2, ..., A_r$ of Θ, the vector $(G(A_1), ..., G(A_r))$ is random. We say G is *Dirichlet process* distributed with base distribution H and concentration parameter α, written $G \sim DP(\alpha, H)$, if

$$(G(A_1), ..., G(A_r)) \sim \text{Dir}(\alpha H(A_1), ...\alpha H(A_r))$$

for every finite measurable partition $A_1, A_2, ..., A_r$ of Θ.

The base distribution H is basically the mean of the DP. The concentration parameter can be understood as an inverse variance [48].

Property 2.3.13. *Let $G \sim DP(\alpha, H)$, for any measurable set A, we have $\mathbb{E}[G(A)] = H(A)$, Var$[G(A)] = H(A)(1 - H(A))/(\alpha + 1)$.*

The proof of Property 2.3.13 can be referred to [49]. It means that the larger α is, the smaller the variance is, and the DP will concentrate more of its mass around the mean. As $\alpha \to \infty$, we will have $G(A) \to H(A)$ for any measurable A, that is $G \to H$ weakly or pointwise (it is not equivalent to saying that $G \to H$, as draws from a DP will be discrete distributions with probability one, even if H is smooth) [48]. For a random distribution G to be distributed according to a DP, its marginal distribution is Dirichlet distributed, similar to the definition of Gaussian process [50].

$\beta_k \sim Beta(1, \alpha)$	$\theta_k^* \sim H$
$\pi = \beta_k \prod_{l=1}^{k-1}(1 - \beta_k)$	$G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k^*}$

TABLE 2.2: Stick-breaking construction of Dirichlet process.

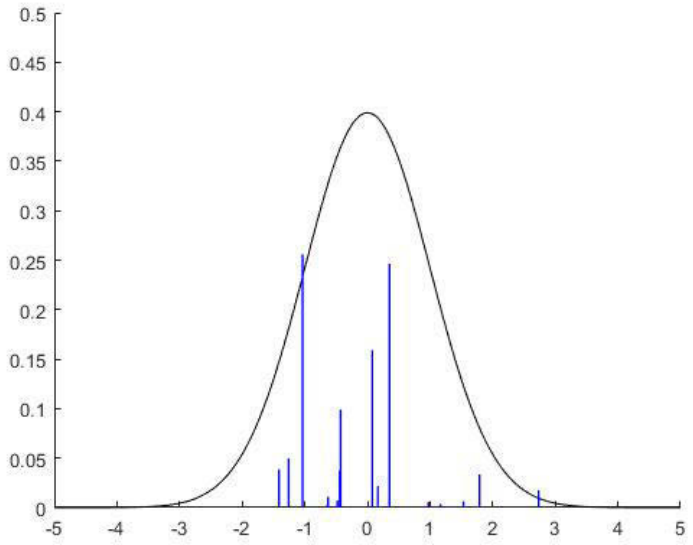


FIGURE 2.5: A sample of Dirichlet process $G \sim DP(\alpha, H)$ with base measure H is $\mathcal{N}(0, 1)$ and $\alpha = 3$.

DP can be constructed by different methods like Stick-breaking, Chinese restaurant process (CRP) and Polya urn model. Table 2.2 presents a stick-breaking construction of the DP [48, 51]. The construction of π can be understood metaphorically as follows. Starting with a stick of length 1, we break it at π_1, assigning π_1 to be the length of stick that we just broke off. Now, recursively break the other portion to obtain π_2, π_3 and so forth shown in Table 2.2 as $\pi = \beta_k \prod_{l=1}^{k-1}(1 - \beta_k)$. Then, $G \sim \text{Dir}(\alpha; H)$. Because of its simplicity, the stick-breaking construction has led to a variety of extensions as well as novel inference techniques for the Dirichlet process.

Fig. 2.5 shows a sample from DP by stick-breaking construction with base measure H is $\mathcal{N}(0, 1)$ and $\alpha = 3$. We have 25 breaking sticks in this example with the largest weight is 0.26.

Proposition 2.3.14 (Posterior distribution of DP)**.** *Let $G \sim DP(\alpha, H)$, given observed values of $\theta_1, \theta_2, ..., \theta_n$, the posterior distribution of G is*

$$(G(A_1), ..., G(A_r))|\theta_1, ..., \theta_n \sim \textit{Dir}(\alpha H(A_1) + n_1, ..., \alpha H(A_r) + n_r). \tag{2.22}$$

Proof. From Bayesian formula, we have

$$p(G(A_1), ..., G(A_r)|\theta_1, ..., \theta_n) \propto p(\theta_1, ..., \theta_n|G(A_1), ..., G(A_r))p(G(A_1), ..., G(A_r)). \tag{2.23}$$

Let $n_k = \sharp\{i : \theta_i \in A_k\}$ be the number of observed values in A_k, we have

$$p(\theta_1, ..., \theta_n|G(A_1), ..., G(A_r)) \propto \text{multinomial}(n_1, ..., n_r). \tag{2.24}$$

By the conjugacy between Dirichlet and multinomial distributions,

$$\begin{aligned} p(G(A_1), ..., G(A_r)|\theta_1, ..., \theta_n) &\propto \text{multinomial}(n_1, ..., n_r)p(G(A_1), ..., G(A_r)) \\ &\propto \text{Dir}(\alpha H(A_1) + n_1, ..., \alpha H(A_r) + n_r). \end{aligned} \tag{2.25}$$

Since $(G(A_1), ..., G(A_r)) \sim \text{Dir}(\alpha H(A_1), ...\alpha H(A_r))$ is true for all finite measurable partition $A_1, ..., A_r$ according to the definition of DP, we have the posterior over G

$$G|\theta_1, ..., \theta_n \sim \text{DP}(\alpha + n, \frac{\alpha}{\alpha + n}H + \frac{n}{\alpha + n}\frac{\sum_{i=1}^{n}\delta_{\theta_i}}{n}) \tag{2.26}$$

is also a DP. □

Based on Eq. 2.26, it is not hard to compute the predictive distribution θ_{n+1} after observing $\theta_1, ..., \theta_n$ by marginalizing out G:

$$p(\theta_{n+1}) = \int_G p(\theta_{n+1}, G|\theta_1, ..., \theta_n) = \mathbb{E}[G(A)|\theta_1, ..., \theta_n] = \frac{\alpha H(A) + \sum_{i=1}^{n}\delta_{\theta_i}(A)}{\alpha + n}. \tag{2.27}$$

Chinese Restaurant Process

We have shown the predictive distribution of DP in Eq. 2.27 by marginalizing out G. Actually, DP can also be represented as Chinese restaurant process (CRP), who is the marginalized DP. It is a distribution over infinite partitions of the integers. This name is derived from a metaphor about a Chinese restaurant.

Let us consider a scenario in a restaurant with an infinite number of tables, and a sequence of customers entering the restaurant and sitting down. The first customer enters and sits at the first

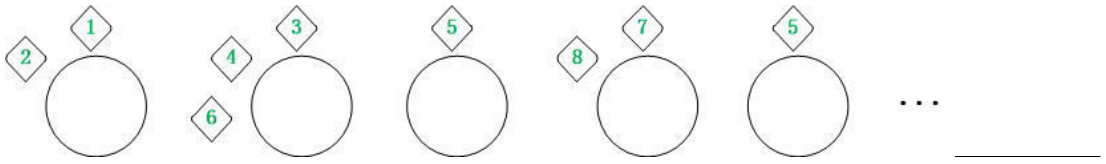


FIGURE 2.6: Example of Chinese restaurant process.

table. The second customer enters and sits at the first table with probability $\frac{1}{1+\alpha}$, and the second table with probability $\frac{\alpha}{1+\alpha}$, where α is a positive real number. When the nth customer enters the restaurant, she sits at each of the occupied tables with proportional to α. At any point in this process, the assignments of customers to tables define a random partition [52]. The scheme of this process is shown as follows.

1. Consider a Chinese restaurant with an unbounded number of tables, $\theta_k, k = 1, ..., \infty$.
2. The first customer sits at table 1.
3. Suppose there are K tables occupied before the ith customer comes, she can sit at

$$\begin{aligned} \text{table } k \quad &\propto \quad \frac{n_k}{i+\alpha-1} \quad \text{if so, set } \phi_i = \theta_k, \\ &\qquad\qquad \text{where } n_k \text{ is the number of customers at table } k; \\ \text{a new table } K+1 \quad &\propto \quad \frac{\alpha}{\alpha+i-1} \quad \text{if so, increase } K \text{ to } K+1, \text{and} \\ &\qquad\qquad \text{draw a new sample } \theta_{K+1} \sim G_0 \text{ and set } \phi_i = \theta_{K+1}. \end{aligned}$$

Then, the random variables $\phi_1, \phi_2, ..., \phi_i$ are distributed according to $G \sim \text{DP}(\alpha, G_0)$. As shown in Fig. 2.6, customers 1 and 2 sit in the first table and customer 9 sits in the fifth table. For a newly coming customer, she has a chance sitting in table $1, 2, ...$ or 5 proportional to the number of customers sitting there and a chance $\frac{\alpha}{\alpha+i-1}$ to sit in a new table.

Remark. As, for $i \geq 1$, the probability that the ith customer takes on a new table is $\alpha/(\alpha+i+1)$, the average number of tables m is:

$$\mathbb{E}(m|n) = \sum_{k=1}^{n} \frac{\alpha}{\alpha+k-1} \in O(\alpha \log n). \tag{2.28}$$

This shows that the number of clusters grows logarithmically with the number of observations. Larger α implies a prior of a larger number of clusters.

Distance Dependent Chinese Restaurant Process

Distance dependent Chinese restaurant process (ddCRP) is a generalization of the Chinese restaurant process that allows for a non-exchangeable distribution on partitions. Rather than representing a partition by customers assigned to tables, the ddCRP models customers linking to other customers. It is based on a random seating assignments of the customers, according to the distances between the data elements. It provides a new tool for flexibly clustering non-exchangeable data [9].

ddCRP alters CRP by modeling customer links to other customers instead of tables. Given a decay function f, sequential distance matrix D and a scaling parameter α, a link can be independently drawn from customer assignments conditioned on distance measurements by the distribution [11]:

$$p(c_i = j) = \begin{cases} f(d_{ij}) & \text{if } i \neq j, \\ \alpha & \text{otherwise.} \end{cases} \tag{2.29}$$

Here, c_i denotes the table assignment of the ith customer, d_{ij} is a distance between data points i and j, and $f(d)$ is called the decay function. The decay function mediates how the distance between two data points affects their probability of connecting to each other, i.e., their probability of belonging to the same cluster.

The difference between the ddCRP and the standard CRP is that in the ddCRP customers sit down with other customers instead of directly at tables. Connected groups of customers sit together at a table only implicitly. ddCRP generalises CRP by introducing a changeable weight on customers $x_{\{i=1,2,\ldots\}}$, and the weight is decided by two things: a distance measure d_{x_i,x_j} between x_i and x_j; and a decay function $f(d)$ which satisfies $f(\infty) = 0$. Moreover, instead of assigning table k to x_i directly, ddCRP assigns x_j to x_i. Such assignments create directed links between customers. The customers that are reachable from each other are assigned to the same table. This yields a non-exchangeable distribution over partitions, since table assignments may change when a new customer assignment is observed. It can be seen that the traditional CRP is an instance of a ddCRP [11].

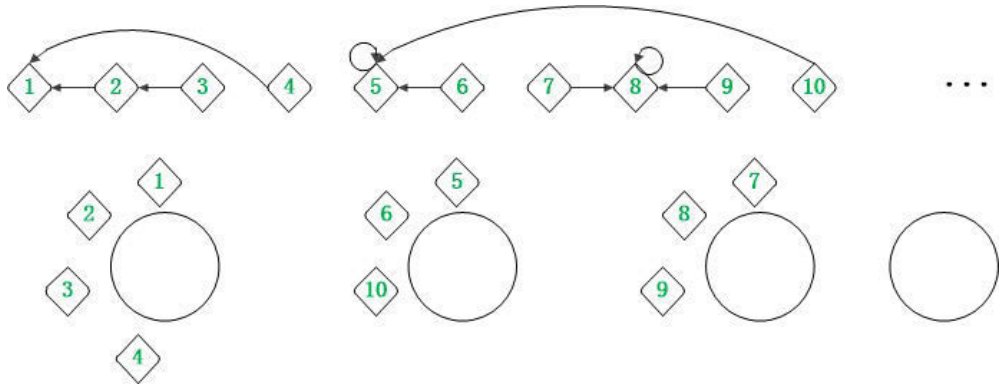


FIGURE 2.7: Example of distance dependent Chinese restaurant process.

Fig. 2.7 illustrates a ddCRP example. Customers sit behind other customers. The distances between the customers determine the seatings, not the order. The circles linked by the customers form tables. Each table means a cluster/group. Here, we can see that, the new customer 10 joins in the second table according to the distance between other customers.

Beta Process

Dirichlet process mixture models are inefficient representations for data if we believe that objects can belong to multiple classes simultaneously. The Beta process is a Bayesian nonparametric prior for sparse collections of binary feature which can be applied for data belong to multiple classes.

Definition 2.18 (Beta process)**.** Let Ω be a measurable space and $\mathcal{B}$ be its σ-algebra. Let H_0 be a continuous probability measure on the space $(\Omega, \mathcal{B})$ and α be a positive scalar. Then, for all disjoint, infinitesimal partitions, $\{B_1, B_2, ..., B_K\}$, of Ω, the Beta process is generated as follows,

$$H(B_k) \sim \text{Beta}(\alpha H_0(B_k), \alpha(1 - H_0(B_k))), \tag{2.30}$$

with $K \to \infty$ and $H_0(B_k) \to 0$, for $k = 1, ..., K$. This process is denoted as $H \sim BP(\alpha H_0)$

A construction description of the Beta process can be presented as follows [5, 14].

Let H_0 be a continuous measure on the space $(\Omega, \mathcal{B})$ and let α be a positive scalar. The process H is defined as follows,

$$\begin{aligned} H &= \sum_{k=1}^{K} \pi_k \delta_{\theta_k} \\ \pi_k &\sim \text{Beta}(\frac{\alpha\gamma}{K}, \alpha(1-\frac{\gamma}{K})) \\ \theta_k &\sim \frac{1}{\gamma} H_0. \end{aligned} \tag{2.31}$$

Then as $K \to \infty$, $H \to \infty$, H is a Beta process.

Similar to the stick breaking construction for Dirichlet process, Beta process can be constructed as follows.

1. Begin with a stick of unit length.
2. For $k = 1, 2, ...$
   (a) sample a Beta$(\alpha, 1)$ random variable μ_k;
   (b) break off a fraction μ_k of the stick, which is the kth atom size;
   (c) throw away what is left of the stick; and
   (d) recur on the part of the stick that was broken off.

Then, π_k is given by $\pi_k = \prod_{j=1}^{k} \mu_j$.

Note. Here, π_k does not sum to 1 unlike the stick breaking construction in Dirichlet process.

Another construction of Beta process $H \sim BP(\alpha H_0)$ by the underlying poisson process can be presented as follows [53].

$$H = \sum_{i=1}^{\infty} \sum_{j=1}^{C_i} V_{ij}^{(i)} \prod_{l=1}^{i-1} (1 - V_{ij}^{(l)}) \delta_{\theta_{ij}} \tag{2.32}$$

$$C_i \sim \text{Poisson}(\gamma) \tag{2.33}$$

$$V_{ij}^{(l)} \sim \text{Beta}(1, \alpha) \tag{2.34}$$

$$\theta_{ij} \sim \frac{1}{\gamma} H_0, \tag{2.35}$$

where, H_0 is a continuous measure on the space $(\Omega, \mathcal{B})$, $H_0(\Omega) = \gamma$ and α is a positive scalar. Then, $H \sim \text{BP}(\alpha H_0)$.

More construction methods about Beta processes are referred to [44, 54–56].

Lévy Measure of Beta Process

Follow the Lévy-Khintchine theorem where a Lévy process is characterized by its Lévy measure, we can get the Lévy measure of the Beta process $BP(cH_0)$ is [57]:

$$\nu(dw, dp) = cp^{-1}(1-p)^{c-1}dpH_0(dw). \tag{2.36}$$

Similar to the Dirichlet process, sample H directly from the infinite Beta process is difficult. However, a marginalized approach can be derived in the same manner as the corresponding Chinese restaurant process, used for sampling from the Dirichlet process. This approach is called Indian buffet process (IBP). Before presenting IBP, we firstly introduce Bernoulli process.

Bernoulli Process

A Bernoulli process is a finite or infinite sequence of independent random variables $X_1, X_2, ...$, such that [58]:

1. For each i, the value of X_i is either 0 or 1;
2. For all values of i, the probability that $X_j = 1$ is the same value p.

In other words, a Bernoulli process is a sequence of independent identically distributed Bernoulli trials. In probability and statistics, a Bernoulli process is a finite or infinite sequence of binary random variables, so it is a discrete-time stochastic process that takes values 0 or 1. The component Bernoulli variables X_i are identical and independent. Thibaux and Jordan described the Bernoulli process in a completely random measure view as follows [57].

Definition 2.19 (Bernoulli process)**.** Let B be a measure on Ω. We define a *Bernoulli process* with measure B, written $X \sim$ BeP(B), as the Lévy process with Lévy measure

$$\mu(dp, d\omega) = \delta_1(dp)B(d\omega). \tag{2.37}$$

If B is continuous, X is simply a Poisson process with intensity $B := X\sum_{i=1}^{N}\delta_{\omega_i}$ where $N \sim$ Poisson$(B(\Omega))$ and ω_i are independently drawn from the distribution $B/B(\Omega)$. If B is discrete

with the form $B = \sum_i p_i \delta_{\omega_i}$, then $X = \sum_i b_i \delta_{\omega_i}$, where the b_i are independent Bernoulli variables with the probability that $b_i = 1$ equals to p_i. If B is mixed discrete-continuous, X is the sum of the two independent contributions.

In summary, a Bernoulli process is similar to a Poisson process, except that it can give weight at most 1 to singletons, even if the base measure B is discontinuous, for instance, B itself is drawn from a Beta process.

An important property between Bernoulli process and Beta process is the conjugacy between them.

Let $B \sim BP(c, B_0)$, and let $X_i|B \sim \text{BeP}(B)$ for $i = 1, 2, ..., n$ be n independent draws from B. Let $X_{1,2,...,n}$ denote the set of observations $\{X_1, X_2, ..., X_n\}$. The posterior distribution of B after observing $\{X_1, X_2, ..., X_n\}$ is still a Beta process with modified parameters:

$$B|X_{1,...,n} \sim \text{BP}(c + n, \frac{c}{c+n} B_0 + \frac{1}{c+n} \sum_{i=1}^{n} X_i). \tag{2.38}$$

Therefore, the Beta process is conjugate to the Bernoulli process [57].

From Eq.2.31, we can see that Beta process has infinite dimensions as $K \to \infty$. A finite approximation to the Beta process H can be made by simply setting K to a large, but finite number. This derives a Beta-Bernoulli approximation for the Beta process as [5]:

$$\begin{aligned} z_{ik} &\sim \text{Bernoulli}(\pi_k) \\ \pi_k &\sim \text{Beta}(\alpha/K, \beta(K-1)/K) \end{aligned} \tag{2.39}$$

here, matrix Z is drawn from a Bernoulli process parameterized by a Beta process.

Indian Buffet Process

Like the Chinese restaurant process corresponding to Dirichlet process, IBP can be seen as a marginalized Beta process. Firstly, we extend the Beta process defined in Eq. 2.31 to take two scalar parameters, α, β and partition the space Ω into K regions of equal measure, that is

$H(B_k) = 1/K$ for $k = 1, 2, ..., K$. We can get

$$\begin{aligned} H(B) &= \sum_{k=1}^{K} \pi_k \delta_{B_k}(B) \\ \pi_k &\sim \text{Beta}(\alpha/K, \beta(K-1)/K), \end{aligned} \tag{2.40}$$

where $B \in \{B_1, B_2, ..., B_k\}$. Marginalizing the vector π (by the Beta-Bernoulli process approximation and the conjugacy between Beta and Bernoulli process), letting $K \to \infty$ and $\beta = 1$, we can get the Indian buffet process described as follows.

Let there be N customers and infinitely number of different dishes.

1. The first customer chooses $K^{(1)}$ different dishes, where $K^{(1)}$ is distributed according to a Poisson distribution with parameter α.

2. The second customer arrives and chooses to enjoy each of dishes already chosen for the table with probability $1/2$. In addition, the second customer choose $K^{(2)} \sim \text{Poisson}(\alpha/2)$ new dishes.

3. The ith customer arrives and chooses to enjoy each of the dishes already chosen for the table with probability m_{ki}/i, where m_{ki} is the number of customers who have chosen the kth dish before the ith customer. In addition, the ith customer chooses $K^{(i)} \sim \text{Poisson}(\alpha/i)$ new dishes.

After the N steps, one has a $N \times K$ binary matrix Z with an IBP prior that describes the customers choices. This Z is a sample from an Indian buffet process [5, 59].

Fig. 2.8 shows an example of IBP. We can see that the IBP is a nonparametric Bayesian prior over distributed partitions (binary matrix) which assumes that objects belong to a small number of latent classes and the expected number of classes grows with the amount of data. Originally, it defines a prior over binary matrices with an infinite number of columns.

The connection between the IBP and the BP can also be studied through their Lévy measures. The IBP with the prior $\pi_i \sim \text{Beta}(c\frac{\gamma}{N}, c)$ can be regarded as a Lévy process with the Lévy measure given as,

$$\nu_{IBP} = \frac{N}{\gamma}\text{Beta}(c\frac{\gamma}{N}, c)d\pi\mu(d\omega). \tag{2.41}$$

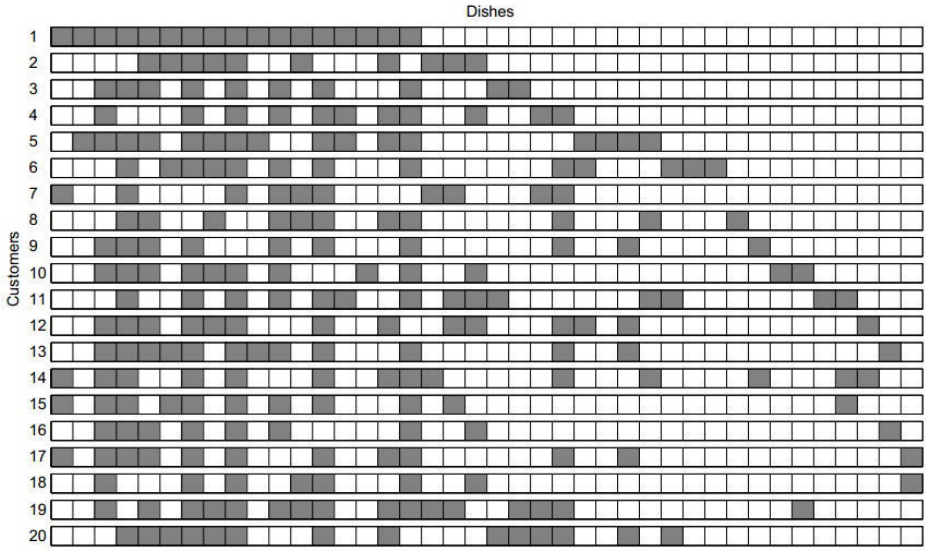


FIGURE 2.8: Example of Indian buffet process [59, 60].

Lévy process	Gaussian variance	Lévy measure	drift	variation	sample path
Brownian motion with drift	$b = \sigma^2$	$\nu \neq 0$	$c = \mu$	infinite	continuous
Compound Poisson process	$b = 0$	$\nu = \lambda f(x)$	$c = 0$	finite	rcll step functions
Poisson process	$b = 0$	$\nu = \lambda\delta(x-1)$	$c = 0$	finite	rcll step functions of step size 1
Gamma process	$b = 0$	$\nu = \beta x^{-1}e^{-\alpha x}$	$c = 0$	finite	rcll
Beta process	$b = 0$	$\nu = cp^{-1}(1-p)^{c-1}$	$c = 0$	finite	rcll

TABLE 2.3: Conclusion of different Lévy processes.

It can be proved that

$$\nu_{IBP} \overset{N\to\infty}{=} \nu, \tag{2.42}$$

where ν is the Lévy measure of the Beta process, which indicates that the Beta process is the limit of the IBP with $N \to \infty$. The detailed proof of this conclusion is presented in the supplementary material of [61].

The inference of Beta process will be presented in the later applications. Here, we summarize the different Lévy processes in Table 2.3 [9].

[9]Here, rcll means right continuous with left limits.

2.4 Markov Chain Monte Carlo (MCMC) methods

Applying probabilistic models to data usually involves integrating a complex, multi-dimensional probability distribution, such as calculating the expectation of a model distribution. Many times, these integrals are not calculable due to no closed-form expression for the integral available using calculus or the high dimensionality of the distribution. Markov Chain Monte Carlo (MCMC) is a method using stochastic sampling routines to approximate complex integrals. This method is composed of two components, the *Markov chain* and *Monte Carlo* integration as MCMC's name indicated.

Monte Carlo integration is a powerful technique that exploits stochastic sampling of the distribution in question, in order to approximate the difficult integration. Monte Carlo integration involves formulating the desired integral as an expectation under the distribution π

$$\mathbb{E}_\pi[f(x)] = \int_{\mathcal{X}} f(x)\pi(x)dx \approx \frac{1}{n}\sum_{1}^{n} f(x_i). \tag{2.43}$$

Thus, we can consider an approximation of the integration in Eq. 2.43, because this approximation has an arbitrary precision for n sufficiently large.

The average of the samples according to the set of independent samples $x_i \sim \pi, i = 1, 2, ..., n$, converges almost surely to the true expectation under distribution π.

Using Monte Carlo method for the integration, it is necessary to have i.i.d samples from the target probability distribution, which may be difficult to access directly. This will resort to the second component of MCMC, the *Markov chain*.

A Markov chain is a random process that transits from one state to another on a state space, where the next state that the chain takes is conditioned on the previous state. It possesses the following three elements [62]:

1. a state space x, which is a set of values that the chain is allowed to take,
2. a transition operator $p(x^{(t+1)}|x^{(t)})$ that defines the probability of moving from state $x^{(t)}$ to $x^{(t+1)}$, and

3. an initial condition distribution $\pi^{(0)}$ which defines the probability of being in any one of the possible states at the initial iteration $t = 0$.

A Markov chain starts at some initial state, which is sampled from $\pi^{(0)}$, then transits from one state to another according to the transition operator $p(x^{(t+1)}|x^{(t)})$. It must possess a property that is usually characterized as *memorylessness*: the probability distribution of the next state depends only on the current state and not on the sequence of events that preceded it. This specific kind of property is called the *Markov property* [63]. Markov chains will take samples from a target probability distribution if they are constructed properly and run for a long time. Therefore, we can apply Monte Carlo integration to perform the approximation of the target distribution by the obtained samples.

MCMC designs a Markov Chain to make it stably converge to the target probability distribution. In order to converge to the target probability distribution, the Markov chain should satisfy some properties.

Detailed Balance

If the transition operator for a Markov chain does not change across transitions, the Markov chain is called time *homogenous*. A nice property of time homogenous Markov chains is that as the chain runs for a long time and $t \to \infty$, the chain will reach an equilibrium that is called the chain's stationary distribution:

$$p(x^{(t+1)}|x^{(t)}) = p(x^{(t)}|x^{(t-1)}). \tag{2.44}$$

The stationary distribution of a Markov chain is important for sampling from probability distributions, a technique that is at the heart of Markov Chain Monte Carlo methods. A necessary condition for drawing from a Markov chain's stationary distribution is the condition known as *reversibility* or *detailed balance*.

In probability theory, a Markov process is said to show detailed balance if the transition rates between each pair of states x and y in the state space obey

$$\pi(x)P(x, y) = \pi(y)P(y, x), \tag{2.45}$$

where $P(x, y)$ is the transition matrix and $\pi(x)$ and $\pi(y)$ are the equilibrium probabilities of being in states x and y, respectively.

Proposition 2.4.1. *Let $K(y|x) = p(x_{n+1} = y|x_n = x)$ be the transition distribution or transition kernel for a given Markov chain. If $K(y|x)$ satisfies detailed balance*

$$\pi(x)P(x, y) = \pi(y)P(y, x). \tag{2.46}$$

Then, the chain defined by this transition kernel has stationary distribution π. A Markov chain satisfying detailed balance is said to be reversible with respect to π.

The proof of this proposition is referred to [6]. From Proposition 2.4.1, we can see that, in order to use Markov chains to sample from a specific target distribution, we have to design the transition operator such that the resulting chain reaches a stationary distribution that matches the target distribution. This is where MCMC methods like the Metropolis sampler, the Metropolis-Hastings sampler, and the Gibbs sampler come to rescue.

2.4.1 MH Sampler

We can use Markov chain to sample from the target probability distribution $p(x)$ from which drawing samples directly is challenging. To do so, as mentioned before, it is necessary to design a transition operator for the Markov chain which makes the chain's stationary distribution converge to the target distribution. The Metropolis-Hastings uses simple heuristics to implement such a transition operator.

The Metropolis-Hastings algorithm provides a generic method for construction an ergodic Markov chain, relying only on a valid proposal distribution $q(\cdot)$ and evaluation of the target distribution $\pi(\cdot)$ up to a normalization constant [6, 7].

Specifically, to draw M samples using the Metropolis-Hasting sampler can be outlined as [64]:

1. initialize t=0;
2. generate an initial state $x^{(0)} \sim \pi^{(0)}$;

3. repeat until $t = M$

   a) set $t = t + 1$;

   b) generate a proposal state x^* from $q(x|x^{(t-1)})$;

   c) calculate the proposal correction factor $c = \frac{q(x^{(t-1)}|x^*)}{q(x^*|x^{(t-1)})}$;

   d) calculate the acceptance probability $\alpha = \min(1, \frac{p(x^*)}{p(x^{(t-1)})\times c})$;

   e) draw a random number u from Unif(0,1); and

   f) if $u \leq \alpha$ accept the proposal state x^* and set $x^{(t)} = x^*$,

   else set $x^{(t)} = x^{(t-1)}$.

Specially, when the proposal distribution is symmetric, the correction factor is equal to one. This gives a special sampler of Metropolis-Hastings algorithm: Metropolis sampler. In order to be able to use an asymmetric proposal distributions, the Metropolis-Hastings algorithm implements an additional correction factor α, defined from the proposal distribution as $c = \frac{q(x^{(t-1)}|x^*)}{q(x^*|x^{(t-1)})}$. The correction factor adjusts the transition operator to ensure that the probability of moving from $x^{(t-1)} \to x^{(t)}$ is equal to the probability of moving from $x^{(t-1)} \to x^{(t)}$, discarding the proposal distribution.

Proposition 2.4.2 (Detailed balance of Metropolis-Hastings sampler)**.** *The Metropolis-Hastings sampler satisfies the detailed balance equation.*

Proof. Let $\pi(\cdot)$ be the target distribution, $q(\cdot)$ be the proposed distribution, and $P(x, y)$ be the transition matrix. We have

$$\begin{aligned} \pi(x)P(x,y) &= \pi(x)q(y|x)\alpha(x,y) \\ &= \pi(x)q(y|x)\text{min}[1, \frac{q(x|y)}{q(y|x)}\frac{\pi(y)}{\pi(x)}] \\ &= \text{min}(\pi(x)q(y|x), \pi(y)q(x|y)) \\ &= \pi(y)P(y,x). \end{aligned} \tag{2.47}$$

□

As α is not always equal to 1, MH samples may be rejected, which will lead to excess computation that is never used. In next section, we will introduce a more efficient algorithm Gibbs sampler.

2.4.2 Gibbs Sampler

The Gibbs sampler, another popular MCMC sampling technique provides a more efficient method by making the rejection rate be zero. Similar to the MH algorithm, the Gibbs sampler also uses component-wise updates. Different from the MH sampler, Gibbs sampler draws from a proposal distribution for each dimension. It simply draws a value for that dimension according to the variable's corresponding conditional distribution. Then, all the values drawn can be accepted.

Given a target distribution $p(\mathbf{x})$, where $\mathbf{x} = (x_1, x_2, ..., x_D)$, the Gibbs sampler is applicable for certain classes of problems, based on two main criteria [65]:

1. be able to get the analytic expression for the *conditional distribution* of each variable in the joint distribution given all other variables in the joint;
2. be able to sample from each conditional distribution.

For the first criterion, if the target distribution $p(x)$ is D-dimensional, we must have D individual expressions for $p(x_i|x_1, x_2, ..., x_{i-1}, ..., x_{i+1}, ..., x_D)$. Having the conditional distribution for each variable means that we do not need a proposal distribution like the MH sampler. Therefore, we can simply sample values from each conditional distribution while keeping all other variables fixed. Different from the MH sampler, we will accept all values that are sampled.

The Gibbs sampling procedure is outlined as follows.

1. initialize $t = 0$;
2. generate an initial state $x^{(0)} = [x_1^{(0)}, x_2^{(0)}, ..., x_d^{(0)}]$;
3. repeat until t=M
   a) set $t = t + 1$;
   b) for each dimension $i = 1, 2, ..., D$,
   draw x_i from $p(x_i|x_1, x_2, ..., x_{i-1}, ..., x_{i+1}, ..., x_D)$.

This Markov chain will converge to the target probability distribution $\pi(x_1, ...x_d)$.

Proposition 2.4.3. *The acceptance ratio of a Gibbs sampler is 1.*

Proof. Let $\pi(\cdot)$ be the target distribution, $q(\cdot)$ be the proposed distribution which is equal to the corresponding conditional distribution. Then, the accept ratio satisfies

$$\begin{aligned}\alpha &= \frac{\pi(x')}{\pi(x)}\frac{q(x|x')}{q(x'|x)} \\ &= \frac{\pi(x')\pi(x_i|x'_{-i})}{\pi(x)\pi(x'_i|x_{-i})} \\ &= \frac{\pi(x')\pi(x_i,x'_{-i})\pi(x_{-i})}{\pi(x)\pi(x'_i,x_{-i})\pi(x'_{-i})} \\ &= 1.\end{aligned} \tag{2.48}$$

where, x_{-i} means $(x_1, x_2, ..., x_{i-1}, ..., x_{i+1}, ..., x_D)$. □

The Gibbs sampler is a popular MCMC method for sampling from complex, multivariate probability distributions. However, It cannot be used for general sampling problems as the first criterion presented before. For many target distributions, it may difficult or impossible to satisfy the first criterion that need closed-form expression for all conditional distributions. In other scenarios, analytic expressions may exist for all conditionals but it may be difficult to sample from any or all of the conditional distributions. Therefore, Gibbs samplers are only suitable for Bayesian methods where models are devised in such a way that conditional expressions for all model variables are easily obtained. Another issue about Gibbs sampling, like many MCMC techniques, suffers from "slow mixing".

More advanced techniques about MCMC are referred to [66–68].

Chapter 3

Face Hallucination Based on Spatial-Distance-Dependent Nonparametric Bayesian Learning

3.1 Introduction

Face image-based techniques have been well developed and investigated in recent years. These techniques have been widely used in many applications such as face recognition, video surveillance, facial expression recognition, image enhancement, image compression, and so on. However, due to the limitations of capturing systems and the changes of environment, human face images captured are very often of low resolution. The poor quality of face images has adverse effect on the performance of computer vision and pattern recognition applications. To solve the problem, it is necessary to render a high-resolution (HR) face image from the corresponding low-resolution (LR) one. This technique is named face hallucination or face super-resolution (SR) [69, 70].

The major difference between face hallucination and the general super-resolution problem is that the face images have regular structures and textures. Compared with the general super-resolution problem, face hallucination is challenging because people are sensitive to the changes in appearances and the quality of human face images. Small deviations might significantly affect

human perception, whereas for super-resolution of generic images, such as buildings, plants, etc., the errors can be more tolerant [71]. Another challenge of hallucinating face images is that faces may be in complex conditions, such as under variations of illumination, pose, and expression. Furthermore, it is difficult to align faces in LR images [69, 71].

Different methods have been researched for face hallucination. One simple way is through the use of interpolation with a base function or through interpolating a kernel function to produce a higher density of pixels in a processed image [72]. Because of the simplicity of interpolation, it is applied in those applications with low requirements. However, this parametric method is often unable to interpolate details well, such as texture and corner-like local regions [71, 73].

Compared with the interpolation methods, using edge-statistical information can well reconstruct edge and corner areas. Fattal and Raanan [74] imposed edge statistics for image upsampling. Sun et al. [75] proposed an image hallucination method by using edge and primal sketch priors. Gradient profile prior was used to enhance the quality of the hallucinated HR image [76, 77]. The major drawback of the learning approaches using edge priors is that they focus on preserving edges so the performance in relatively smooth regions is mediocre as discussed in [2,17].

Example-based super-resolution schemes have proven to be able to reconstruct significantly finer details from a LR image compared with the interpolation-based schemes [73]. The general idea of example-based approaches is to learn the statistical correlation between pairs of LR and HR images from a face dataset. The learned correlation is then applied to an input LR image to reconstruct the corresponding HR image [71]. Different methods have been studied to learn the mapping relationship between LR and HR images [69, 78], such as

1. Sparse representation-based approaches [79, 80];
2. Subspace learning approaches, including locally linear embedding and linear subspace learning-based approaches [72]; and
3. Bayesian inference method: learning priors from numerous feature vectors to generate a function, mapping features from LR images to HR images [73, 81].

Usually, example-based methods require computationally expensive processes in extracting complex features or searching exemplars [69]. In [82], a divide-and-conquer algorithm was proposed

by using k-means to learn K clusters. Each cluster can be viewed as a set of anchor points to represent the feature space of natural image patches for super resolution. For different data sets or applications, K should be set adaptively. The training can be done off-line, and the reconstruction is performed by the efficient linear regression method.

Performance of learning-based SR methods heavily depends on the similarity between the training and the testing images to query input LR face images. The quality of the edges in a reconstructed HR image can be significantly degraded when the edges in training images cannot be matched or aligned well with the corresponding input image. Recent research has presented that the structural constraint can be applied to improve the results of face hallucination. For example, Markov random fields can be used to reduce the ambiguity between LR and HR images by learning the statistical relationship between a global face image and its local features [71]. The structure or position information about face images can be used to improve the face hallucination performance. Yang et al. [83] used a landmark localization method to estimate and align facial features for hallucination. Jung et al. [84] proposed a position-patch-based face hallucination method using convex optimization. However, Yang's method is strongly dependent on the results of landmark localization, while Jung's method requires the faces to be aligned accurately.

Inspired by [82] and recent development of nonparametric Bayesian methods, we propose a novel framework in this chapter. A nonparametric Bayesian method, namely distance dependent Chinese restaurant process (ddCRP), is employed to model the spatial dependencies between the patches in LR and HR face images and to learn the corresponding mapping matrices for the patches. Without requesting an accurate alignment, the spatial constraint provided by ddCRP can produce more details in the final reconstructed HR image as face images are characterized by similar spatial structures.

The remainder of this chapter is organized as follows. Section 2 reviews the Bayesian image super-resolution model and the background of ddCRP. In Section 3, we present the details of our proposed model and the inference process. In Section 4, implementation details are described, and experimental results are presented to show the performance of our method. Finally, in Section 5, we draw a conclusion and discuss the relationship between the constraint of distance dependence in ddCRP and MRF.

3.2 Bayesian Image Super-resolution Model

A common image degradation model can be presented in a Bayesian super-resolution framework [81, 85, 86]. According to this Bayesian model, an LR image is assumed to be generated from its HR counterpart through blurring, sub-sampling and including additive noise, independently to others. Example-based methods usually use image patches for super resolution. For an LR image X divided into N patches, the kth patch X_k is assumed to be generated from the HR image patch Y_k as follows.

$$X_k = W_k Y_k + \epsilon_k, k = 1, 2, ..., N, \tag{3.1}$$

where ϵ_k is a matrix of i.i.d Gaussian noise and W_k is a degradation matrix related to down-sampling and blurring in a Gaussian form.

According to this model, the reconstruction process is

$$Y_k = A_k X_k + \mathcal{E}_k, k = 1, 2, ..., N, \tag{3.2}$$

where $\mathcal{E}_k$ is the reconstruction error with a residual noise covariance matrix Σ_k, and A_k is the reconstruction mapping matrix.

The optimal HR image is reconstructed by maximizing $p(Y_k|X_k)$, i.e,

$$Y_{k_{MAP}} = \underset{Y_k}{\operatorname{argmax}}(p(Y_k|X_k, A_k, \Sigma_k)). \tag{3.3}$$

In this work, we place a conjugate matrix normal-inverse-Wishart prior on the parameters A_k, Σ_k.

$$\begin{aligned} \Sigma_k &\sim \mathcal{IW}(n_0, S_0) \\ A_k|\Sigma_k &\sim \mathcal{MN}(M, \Sigma_k, I). \end{aligned} \tag{3.4}$$

where I is an identity matrix.

A matrix $A \in R^{m \times n}$ has a matrix-normal distribution $\mathcal{MN}(A; M, V, U)$ if

$$p(A) = \frac{|U|^{\frac{d}{2}}}{|2\pi V|^{\frac{m}{2}}} e^{-\frac{1}{2} tr((A-M)^T V^{-1}(A-M)U)}, \tag{3.5}$$

where M is the mean matrix, and V and U^{-1} are the covariance matrices along the rows and columns, respectively.

The reconstruction matrix A_k and covariance Σ_k are unknown, which can be learned by using the Bayesian model. We learn A_k and covariance Σ_k from a cluster of patches similar to X_k and Y_k.

In the testing stage, when A_k and Σ_k are estimated, given the LR image patch X_k, the corresponding HR image patch Y_k can be easily reconstructed as follows:

$$\widehat{Y_k} = A_k X_k. \tag{3.6}$$

3.3 Proposed method

Dirichlet process (DP), as a nonparametric Bayesian method, provides a valuable suite of flexible clustering algorithms for high-dimensional data analysis. It has been extensively used in computer vision and pattern recognition areas such as image segmentation, text modelling, computational biology, and so on [9–11, 87]. DP can be described via the Chinese restaurant process (CRP). The probability of a customer sitting at a table is computed from the number of other customers already sitting at that table. Despite the success of the traditional CRP, it ignores the spatial distance between data elements. Distance dependent Chinese restaurant process, which is based on the random seating assignment of the customers, considers the distances between the data elements. It provides a new tool for flexibly clustering non-exchangeable data.

ddCRP alters CRP by modeling customer links to other customers instead of tables. Given a decay function f, sequential distance matrix D and a scaling parameter α, a link can be independently drawn from customer assignments conditioned on distance measurements by the distribution [11]

$$p(c_i = j) = \begin{cases} f(d_{ij}) & \text{if } i \neq j, \\ \alpha & \text{otherwise.} \end{cases} \tag{3.7}$$

In our application, d_{ij} is an externally specified distance between image patches i and j. α determines the probability that a patch links to itself rather than to another patch. The decay function f decides how the distance between two patches affects their probability of connecting to each other.

More details of ddCRP are referred to [11].

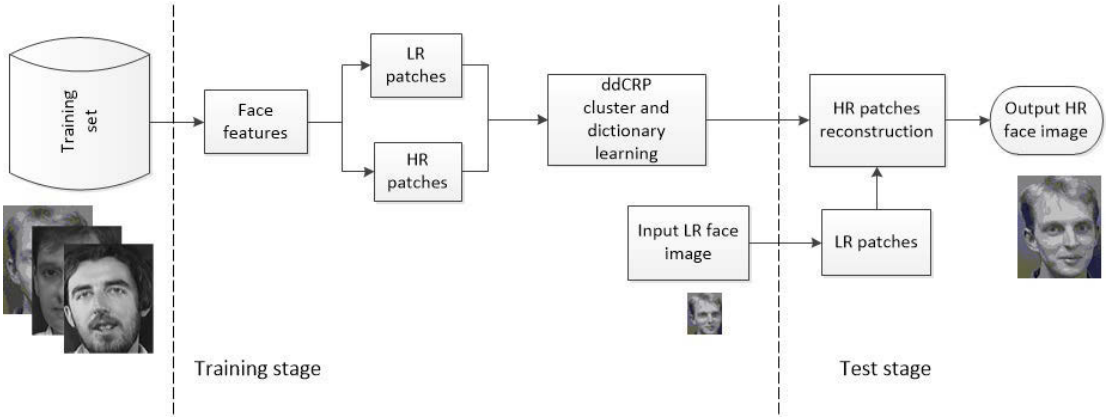


FIGURE 3.1: The framework of our proposed method.

Figure 3.1 shows the framework of the proposed face hallucination method. Firstly, we collect a large set of LR and HR paired patches from a training set of human face images. The intensity values of a set of LR patches are subtracted, by their respective means, to form the feature vectors. We learn the cluster centers of the feature vectors using ddCRP, and the functions mapping the LR patches to the corresponding HR patches in each cluster. After the coefficients of mapping functions on each cluster are learned, in the test stage the HR patches of an input LR face image can be simply generated by using Eq. 3.6. Finally, the HR face image is reconstructed from the HR patches, and the zooming LR mean is added.

3.3.1 Training Stage

At the training stage, given a set of HR face images, the corresponding LR images are generated by using a Gaussian kernel followed by down-sampling [82, 88]. We randomly extract a large set of HR and LR patches from the HR and LR image pairs to form a training set.

We expect that the patches in the same cluster have similar distribution and are constrained by their positions in the face images. This does not require the face images to be strictly aligned.

In ddCRP, two customers (or patches in face hallucination) are placed in the same table (or cluster) if one can reach the other by traversing the customer (or patch) assignments. Let us consider a collection of N LR and HR image patch pairs in the training dataset, denoted as $X = [X_1, ..., X_N]$ and $Y = [Y_1, ..., Y_N]$, respectively. The full generative process for the observed patches $X_{1:N}$ is described as follows.

1. For customer/patch i, $i \in \{1, ..., N\}$, sample the assignment $c_i \sim ddCRP$. This determines the table/cluster assignment z.
2. For table/cluster k, $k \in \{1, ...\}$, sample parameter $\phi \sim G_0$.
3. Given ϕ and the assignment c_i, independently sample the patches.

We place a conjugate normal-inverse-Wishart prior on the parameters A, and Σ of the patch distribution G_0. Assuming that the assignment of patch i is $z(c_i) = k$, given A_k and Σ_k, Y_i can be sampled from $Y_i \sim \mathcal{MN}(A_k X_i, \Sigma_k, I)$. The full model of the observed patches is:

$$\begin{aligned} p(Y, c, A, \Sigma | X, D, \alpha, f) &= p(c|D, f, \alpha, \eta) \\ &\prod_{i=1}^{N} \mathcal{N}(Y_i | X_i, A_{z(c_i)}, \Sigma_{z(c_i)}) [\prod_{k=1}^{z(c_i)} p(A_k, \Sigma_k, \eta)], \end{aligned} \tag{3.8}$$

where η represents a hyperparameter. This model also determines the number of clusters for the training dataset.

Posterior inference is the central computational focus for analyzing the data X and Y and for learning the parameters in our nonparametric Bayesian face hallucination model. Here, we use Gibbs sampler to infer the proposed model.

Gibbs sampling is a Markov chain Monte Carlo (MCMC) algorithm for obtaining a sequence of samples which are approximated from the target probability distribution when direct sampling is difficult. A sequence of samples is drawn from the conditional posterior densities of the model parameters, and it will converge to a sample from the joint posterior.

Precisely, the assignment of each patch can be described as follows.

$$p(c_i|\mathbf{c}_i^-, X, Y, A, D, f, \alpha, \eta) \propto \begin{cases} p(c_i|X, Y, A, D, f, \alpha)\Delta(X, Y, A, z(c), \eta) \\ \quad \text{if } c_i \text{ links } k_1 \text{ and } k_2 \\ p(c_i|D, f, \alpha) \quad \text{otherwise.} \end{cases} \tag{3.9}$$

In Eq. 3.9, $\mathbf{c}_i^-$ is the assignment set excluding c_i (remove the patch link c_i from the current configuration);

$$\Delta(X, Y, A, z, \phi) = \frac{P1}{P2P3},$$

$$P1 = p(Y_{z(c_{1:N})=k}|X_{z(c_{1:N})=k}, A_k, \eta),$$

$$P2 = p(Y_{z(c_{1:N})=k_1}|X_{z(c_{1:N})=k_1}, A_{k1}, \eta),$$

$$P3 = p(Y_{z(c_{1:N})=k_2}|X_{z(c_{1:N})=k_2}, A_{k2}, \eta).$$

where k is a cluster index, and k_1 and k_2 are the indices of the clusters that together form the cluster with index k, when c_i makes the clusters corresponding k_1 and k_2 join together.

To compute $p(Y|X, A, \eta)$, firstly, based on the conjugate prior of A and Σ, we have [87, 89]

$$\begin{aligned} p(Y|X, \Sigma) &= \int p(Y, A|X, \Sigma)dA \\ &= \frac{|U|^{d/2}}{|2\pi\Sigma|^{N/2}|S_{xx}|^{d/2}} \exp(-\frac{1}{2}tr(\Sigma^{-1}S_{y|x})) \\ S_{xx} &= XX^T + U \\ S_{yx} &= YX^T + MU \\ S_{y|x} &= YY^T + MUM^T - S_{yx}(S_{xx})^{-1}S_{yx}^T. \end{aligned} \tag{3.10}$$

Then, the part likelihood is:

$$\begin{aligned} p(Y|X) &= \int p(Y|X, \Sigma)p(\Sigma|n_0, S_0)d\Sigma \\ &= \frac{|K|^{\frac{3}{2}}|S_0|^{\frac{n_0}{2}}\Gamma_3(\frac{N+n_0}{2})}{\pi^{\frac{3N}{2}}|S_{xx}|^{\frac{3}{2}}|S_0 + S_{y|x}|^{\frac{N+n_0}{2}}\Gamma_3(\frac{n_0}{2})}. \end{aligned} \tag{3.11}$$

Eq. 3.9 guarantees that patches with similar texture, constrained by their positions, are assigned to the same cluster.

Sample A:

Given the observed data X_k and Y_k, the posterior of the reconstruction matrix A_k and the noise covariance Σ_k is given as follows.

$$\begin{aligned} &p(A_k, \Sigma_k | X_k, Y_k) = \\ &p(A_k | X_k, Y_k, \Sigma_k) \mathcal{IW}(\Sigma_k | N_k + n_0, S_{y|x} + S_0), \end{aligned} \tag{3.12}$$

where N_k is the number of patches in cluster k.

The prior of the noise covariance is conjugate to Matrix normal distribution, so we can integrate out Σ_k [87]:

$$\begin{aligned} &p(A_k | X_k, Y_k, \eta) = \\ &\int p(A_k | X_k, Y_k, \Sigma_k) \mathcal{IW}(\Sigma_k | N_k + n_0, S_{y|x} + S_0) d\Sigma_k \\ &= \mathcal{MT}(A_k | N_k + n_0, S_{yx} S_{xx}^{-1}, S_{xx}, S_{y|x} + S_0), \end{aligned} \tag{3.13}$$

where $\mathcal{MT}(\cdot)$ is a matrix-t distribution.

Sample Σ:

By Eq. 3.13, it is easy to see that the posterior distribution of Σ_k is an inverse-Wishart marginal posterior distribution [89], as follows.

$$\begin{aligned} p(\Sigma_k | Y_k, X_k) \propto & p(Y_k, X_k | \Sigma_k |) p(\Sigma_k) \\ \propto & IW(N_k + n_0, S^k_{y|x+S_0}). \end{aligned} \tag{3.14}$$

Based on the above analysis, we can approximate the posterior using a Gibbs sampler, which iteratively draws the parameters from the conditional distribution. Then, the proposed model can be sampled by Algorithm 1, as shown in the following.

Algorithm 1 Gibbs sampler of the proposed method

Input:

- $Y_1, ..., Y_n, X_1, ..., X_n$,
- Position matrix D.

Initialize:

Generate α, s_0, n_0 and $\theta \sim G_0(\theta)$ and set $\theta_i = \theta$ for $i = 1, 2, ..., n$.

Repeat: In each sample iteration

1. For patch $i = 1, 2, ..., N$:
   (a) Compute the assignment of each patch by Eq. 3.9;
   (b) Remove the empty clusters;
   (c) Sample A by Eq. 3.13;
   (d) Sample Σ by Eq. 3.14;
2. For each cluster $k = 1, ..., N_c$:
   Update the cluster center.

Output:

- Reconstruct Matrices A, Σ and cluster centers.

3.3.2 Testing Stage

At the testing stage, the input LR image is divided into patches. The features of each LR patch are computed, and the corresponding closest cluster center and the dictionary A are identified. The predicted HR patch is then reconstructed through linear regression based on the learned mapping coefficients obtained by using Eq. 3.4 , and then the LR patch mean is added to the HR features, as follows.

$$Y_k = A_k \cdot f(X_k) + \text{mean}(X_k), \tag{3.15}$$

where $f(\cdot)$ is a gradient feature extraction function of the input LR image. Finally, the HR face image is reconstructed by averaging the overlapped areas of the HR patches.

3.4 Experiments

To illustrate the performance of the proposed method, we evaluate our algorithm on ORL face database [90] and Yale face database [91] with different zooming factors.

Images	Bicubic	Chang's method	Jian's method	Yang's method	proposed method
1	27.6764	27.1460	27.7070	27.8224	**28.1955**
2	27.1475	27.4362	27.1869	**27.9673**	27.8206
3	27.3182	28.1702	27.3340	27.9821	**27.9968**
4	28.7479	**29.9317**	28.7732	29.3229	29.5897
5	28.5319	28.5361	28.5627	29.0574	**29.1382**

TABLE 3.1: PSNR performance of different algorithms on ORL database.

3.4.1 Face Hallucination on ORL Database

The ORL face database is composed of 400 images of 40 persons. Each person has ten different images taken at different times, lighting conditions, facial expressions and facial details. The size of each image is 92×112 pixels, with 256 gray levels per pixel.

In our experiments, the training set contains 200 face images, which are randomly selected from the ORL database. The remaining images are used for testing. Without loss of generality, we magnify the input face image with a factor of 3. In other words, the original images form the HR image dataset, which are down-sampled with a factor of 3 to form the LR image dataset. 10,000 patches are generated from the LR and HR training sets. As suggested by [82], the LR patch size is 7×7 pixels while the size of each HR patch is 11×11 pixels.

The hyperparameters that regularize the ddCRP prior can be specified based on the properties of the face image patches and the training set. We use the Euclidean distance for the matrix D shown in Eq. 3.4. The decay function f is set to 1 when the distance is larger than 2 pixels, with the self-connection parameter $\alpha = 10^{-6}$, which is set by experiences. For the hyperparameter of the matrix-normal-inverse-Wishart prior, we set the degree of freedom $n_0 = 50$, a value which makes the prior variance nearly as large as possible while ensuring that the mean remains finite.

The samplers present rapid mixing and often stabilize within 20 iterations. The similar rapid mixing has been observed in other ddCRP applications [9, 11]. Here, we run the sampler for 100 times and obtain 16 clusters with more than 150 patches. The converging time depends on the distance defined previously and the size of the training set.

Images	Bicubic	Chang's method	Jian's method	Yang's method	proposed method
1	0.8721	0.8625	0.8727	0.8834	**0.8877**
2	0.8190	0.8303	0.8203	**0.8435**	0.8430
3	0.7847	0.8118	0.7857	0.8167	**0.8239**
4	0.8407	0.8624	0.8411	0.8566	**0.8673**
5	0.8667	0.8703	0.8677	0.8854	**0.8893**

TABLE 3.2: SSIM performance of different algorithms on ORL database.

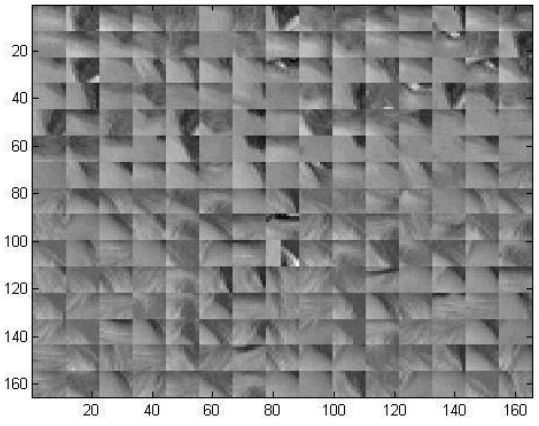


FIGURE 3.2: Patches learned in an example cluster.

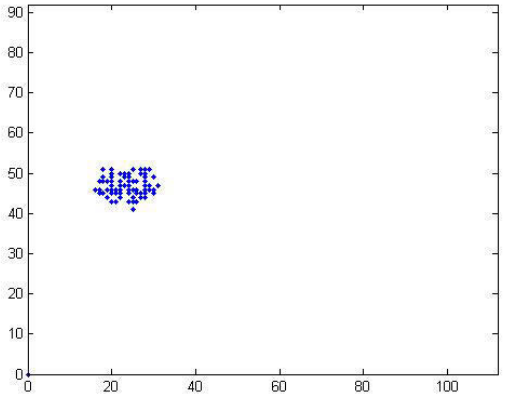


FIGURE 3.3: Patches positions constrained in the example cluster.

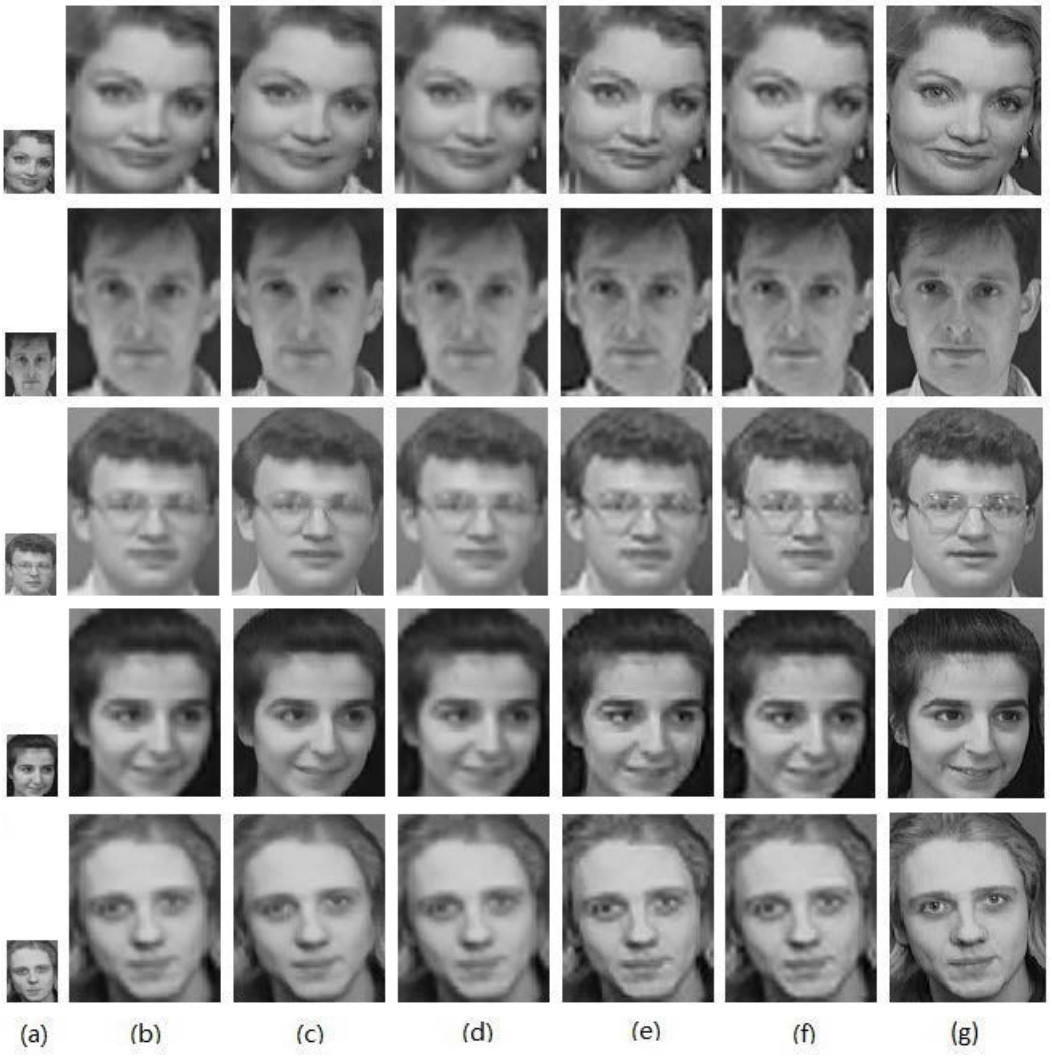


FIGURE 3.4: Face hallucination results on the ORL database with different methods: (a) the input LR faces, (b) Bicubic interpolation, (c) Chang's method, (d) Sun's method, (e) Yang's method, (f) our proposed method, and (g) the original HR faces.

Figure 3.2 presents the LR patches in one cluster learned from the training dataset. It can be seen that they have similar textures. Figure 3.3 presents the corresponding spatial center positions of the patches shown in Figure 3.2.

We also compare our method with some typical face hallucination methods. PSNR and SSIM are used as the objective measurements of image quality. The results based on different methods are shown in Tables 3.1 and 3.2.

Figure 3.4 presents the visual results of 4 face images based on the different methods. Column (a) shows the original LR face images. The second column (b) presents the results based on

Images	Bicubic	Chang's method	Jian's method	Yang's method	proposed method
1	27.0842	28.0617	27.2738	**28.2065**	28.0767
2	26.2234	26.2402	26.2820	26.5094	**26.5945**
3	26.2025	27.0032	26.4725	27.3066	**27.4979**
4	27.0842	28.0617	27.2738	27.8767	**28.1320**

TABLE 3.3: PSNR performance of different algorithms on Yale database.

Images	Bicubic	Chang's method	Jian's method	Yang's method	proposed method
1	0.8584	0.8405	0.8596	**0.8894**	0.8827
2	0.8248	0.8232	0.8269	0.8352	**0.8452**
3	0.8097	0.7857	0.8105	0.8335	**0.8343**
4	0.8584	0.8405	0.8595	0.8554	**0.8652**

TABLE 3.4: SSIM performance of different algorithms on Yale database.

Bicubic interpolation. The results of Chang's method based on locally linear embedding (LLE) [78], Sun's method using the edge statistic information Gradient Profile Prior [75], and Yang's method [79] by sparse representation are shown in columns (c), (d), and (e), respectively.

3.4.2 Face Hallucination on Yale Database

The Yale Face Database contains 165 grayscale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised, and wink. In our experiments, the training set contains 100 face images which are randomly selected from the database. The remaining images are used for testing. We crop the images into size of 240×204 pixels, with some background included. Similar to the experiments on the ORL database, 10,000 patches pairs are generated in the LR and HR training set. The size of the face images in the Yale database is larger than those in the ORL database. With this database, the input LR face images are magnified with a zooming factor of 4. The LR patch size is 7×7 pixels and the HR patch is set to 12×12 pixels. The other hyperparameters used are the same as those used in the ORL experiments.

Figure 3.5 presents the visual results with 4 face images based on the different methods. Tables 3.3 and 3.4 show the PSNR results and SSIM of the results for the different methods.

FIGURE 3.5: Face hallucination results on the Yale database with different methods: (a) the input LR faces, (b)Bicubic interpolation, (c) Chang's method, (d) Sun's method, (e) Yang's method, (f) our proposed method, and (g) the original HR faces.

3.4.3 Performance Analysis

The results from the Bicubic method are the smoothest. As with other interpolation methods, it struggles to reconstruct high-frequency details of the HR images like edges and corners. The neighbor embedding method assumes that the local geometry of LR image patches is similar to that of the HR counterparts. Their performance suffers from inappropriate choices of features, neighborhood sizes and training patches. The reconstruction of high-frequency areas is also mediocre in our experiments. Jian's method using gradient profile prior gives plausible performance in the edge and corner areas when zooming factor is not large (less than 4). It can be seen that Yang's method can provide plausible HR facial images with sharp edges and corners. However, some of the patches may be badly matched or conflict with adjacent ones. These

errors are typical drawbacks with most patch-based SR methods. Furthermore, patch-based SR usually requires a large number of image-patch pairs for learning. When the training dataset size is limited, those drawbacks are more obvious.

Compared with other parch-based learning methods, our proposed method keeps better continuities with adjacent ones without loss of the reconstruction performance of the edge and corner areas. The spatial constraint in our method prevents producing artifacts and discontinuities in the reconstruction results, especially when the train dataset size is not large as in our experiments using only 10,000 patches.

3.5 Conclusion

In this chapter, we have presented a novel example-based face hallucination method based on a nonparametric Bayesian model. According to the assumption that all face images have similar local pixel structures, we have clustered the LR face patches using ddCRP, and learned the mapping coefficients for each cluster. We have used the LR input face to search a database to find sample face patches that best match the input face. The HR image has been reconstructed from a linear regression by the learned mapping matrix. Experimental results have shown that the proposed method performs well in terms of both reconstruction error and visual quality.

In our algorithm, the spatial distance constraint is employed to aid the learning of cluster centers. The patches learned in a cluster are not only similar in texture, but also adjacent in locations. This constraint between the patches can be considered as an extension of the constraint in the Markov Random Fields (MRF). It is able to keep better continuities in the face hallucination result. The PSNR and SSIM results of the experiments show that our method can achieve competitive performance for face hallucination. Moreover, MRF is usually applied to the neighborhood of the patches in the same image. The spatial distance constraint can be applied to the patches from different images. Therefore, this is a more flexible constraint to describe the neighborhood of face image pixels.

Chapter 4

A Graph-regularized Nonparametric Bayesian Approach to Sparse Nonnegative Matrix Factorization

4.1 Introduction

Nonnegative Matrix Factorization (NMF) is a dimension reduction method for factorizing a matrix as a product of two matrices, in which all elements are nonnegative. It has been widely used in various areas including clustering, data mining, machine learning, pattern recognition, computer vision and so forth. Compared with other unsupervised learning algorithms such as principal components analysis (PCA), independent component analysis (ICA), vector quantization (VQ), etc., the factors of NMF give a better natural interpretation as the nonnegative constraints [92–95]. For example, the parts contained in the objects can be learned from the factors of lower rank approximation by using NMF. Each object is explained by an additive linear combination of intrinsic 'parts' [93, 96]. Many physical signals, such as image intensities, amplitude spectral and text documents, are naturally represented by nonnegative values [97]. Therefore, NMF is a powerful method for analyzing such data. An NMF problem can be formulated as follows [98]:

NMF Problem. Given a nonnegative matrix $\mathbf{X} \in R^{m \times n}$ and a positive integer $k < min(m, n)$, find nonnegative matrices $\mathbf{A} \in R^{m \times k}$ and $\mathbf{B} \in R^{k \times n}$ to minimize the functional

$$f(\mathbf{A}, \mathbf{B}) = \frac{1}{2} \parallel \mathbf{X} - \mathbf{AB} \parallel_F^2, \tag{4.1}$$

where $\parallel \cdot \parallel_F$ is assumed to be a matrix norm .

Usually, the columns of $\mathbf{A}$ gain the interpretation of basis vectors, while $\mathbf{B}$ is called a coefficient matrix. The product $\mathbf{AB}$ is called a nonnegative matrix factorization of $\mathbf{X}$ [99]. When the inner dimension of the product is less than the rank of $\mathbf{X}$, then the product is only an approximation of $\mathbf{X}$. Therefore, the factorization is sometimes referred to as the Approximative Non-Negative Matrix Factorization.

Different efforts have been made to solve Equation 4.1. The minimization problem of Equation 4.1 is convex in $\mathbf{A}$ and $\mathbf{B}$ separately. However, it is not convex in both simultaneously. One of the most popular approaches to solve it is the multiplicative update by a gradient decent algorithm, proposed by Lee and Seung [93]. It is simple to implement and often yields good results. In each iteration of this method, the elements of $\mathbf{A}$ and $\mathbf{B}$ are multiplied by certain factors. As the zero elements are not updated, all components of $\mathbf{A}$ and $\mathbf{B}$ are strictly positive for all iterations. Some other methods such as conjugate gradient have been developed later for faster convergence [100]. One typical method is the projected gradient optimization method proposed by Lin and Chih-Jen [100] which is computationally competitive and appears to have better convergence properties than the standard (multiplicative update rule) approach.

These gradient-based algorithms focus on speeding up the convergence. Several authors have noted the shortcomings of the classical NMF that does not always give good results for presentation parts features [99]. Some efforts have been made to enhance the quality of the NMF by adding further constraints on the factors, such as smoothness, sparsity and spatial localization [95, 97]. Piper and Pauce [101] employed regularization on the decomposition. Chen and Cichocki [102] and Bertin et al. [103] used smoothness constraints to improve the analysis of the data for particular applications. Hoyer [95] employed sparsity constraints on either factor $\mathbf{W}$ or $\mathbf{H}$ to improve the local rather than the global representation of data. The extended NMF by sparseness control allows us to discover better parts-based representations of the data than basic NMF. However, it is difficult to control the sparse degree for a good representation.

Most of the above mentioned research on NMF use numerical algorithms for learning the optimal nonnegative factors from data. The problem of NMF can also be interpreted in a Bayesian framework based on the distribution of the matrix $\mathbf{X}$. These methods have the advantages that the underlying assumptions in the model are made explicit and some good properties are presented in the factors. Schmidt and Winther [104] built a Bayesian frame work for NMF by using exponential priors on the factors. Moussaoui et al. [105] used Gamma densities as factors priors. Both of them developed a Gibbs sampler to infer the parameters. Schmidt and Laurberg [97] employed a Gaussian process for the prior of the factors which agreed with the prior knowledge of the factors' distribution, such as sparseness, smoothness and symmetries. However, these Bayesian methods do not have the sparse feature.

In this chapter, we discuss NMF in a graph-regularized nonparametric Bayesian framework. We assume that good basic vectors learned from the training data by the NMF methods should respect the structure of the dataset. A natural idea is that if two data $\mathbf{x}_i$ and $\mathbf{x}_j$ are close to each other in the Euclidean space, they should have similar representation coefficients $\mathbf{z}_i$ and $\mathbf{z}_j$ based on the learned non-negative basic vectors. This idea can be seen as a local invariance constraint which plays an essential role in various kinds of algorithms including semi-supervised learning, matrix completion and dimensionality reduction algorithms [106]. Experimental results present that the proposed method is able to give a good sparse NMF and better reflect the structure of the data.

The rest of this chapter is organized as follows. In Section 2, the classical graph construction methods and nonparametric Bayesian model for NMF are introduced. Section 3 presents the proposed approach for sparse NMF. In Section 4, a Gibbs sampler is designed to build a Markov chain for sampling the joint posterior density of our model. Section 5 presents some factorization results on different data sets. Finally, Section 6 gives the discussion and conclusion.

4.2 Bayesian Modeling

The problem of NMF can be interpreted in a Bayesian framework. The matrix $\mathbf{X}$ is assumed to be generated by the following model:

$$\mathbf{X} = \mathbf{AB} + e, \tag{4.2}$$

where $\mathbf{X} \in R^{M \times N}$, $\mathbf{A} \in R^{M \times k}$, $\mathbf{B} \in R^{k \times N}$, and e is i.i.d white Gaussian noise. All elements in A and B are either positive or zero. The Gaussian likelihood for NMF is given by:

$$p(\mathbf{X} = \mathbf{AB}) = \prod_{i=1}^{M} \prod_{j=1}^{N} \mathcal{N}(X_{i,j} | A_{i:} B_{:j}, \sigma^2), \tag{4.3}$$

where $\mathcal{N}(x|\mu, \sigma^2)$ is Gaussian distribution, and μ is the mean and σ^2 is the noise variance.

We can introduce nonnegative priors such as exponential priors for $\mathbf{A}$ and $\mathbf{B}$ as shown in [104, 105]. However, these methods do not consider the sparse constraints for learning parts in data $\mathbf{X}$. In the following sections, we will introduce a Bayesian nonparametric prior for the sparse NMF problem.

4.2.1 Background of Graph Construction

Graph-based algorithms have been well applied in image analysis and machine learning areas, such as data clustering, subspace learning, semi-supervised learning and so on. A good graph is able to reveal the true intrinsic complexity or dimensionality of the data points, and also capture certain global structures of the data as a whole (i.e. multiple clusters, subspaces, or manifolds) [107, 108].

Usually, graph construction consists two steps: adjacent construction and weight calculation. For the graph construction, there are two popular ways. One is the K-nearest-neighbor (Knn) method, and the other one is the ϵ-ball based method [108]. We select the Knn method in our experiment as it is easier to control the number of neighbors and the graph constructed is symmetric.

For graph weight calculation, there exist several frequently used approaches: Heat Kernel, Inverse Euclidean Distance, Local Linear Reconstruction and so on. These approaches compute the weights for the constructed neighbors by measuring the similarity distance between the adjacent data. The different similarity measures are suitable for different situations. While for image data, the heat kernel presented below is a popular choice [106].

$$W_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{\sigma}} & \text{if } x_i \text{ and } x_j \text{ are neighbors,} \\ 0 & \text{otherwise,} \end{cases} \tag{4.4}$$

where σ is the heat kernel parameter.

Since the weights in our model are only for measuring the closeness, we do not treat the different weighting schemes separately. After the above two steps, we can get a symmetric graph G.

4.2.2 NMF with Beta-Bernoulli Process Prior

Beta process is a nonparametric Bayesian distribution that defines a prior over binary matrices with an infinite number of columns. The two-parameter BP developed in [14] is presented as $BP(a_0, b_0, H)$ with parameter $a_0 > 0$ and $b_0 > 0$, and base measure H_0. A draw $H \sim BP(a_0, b_0, H_0)$ can be presented as

$$\begin{aligned} H(B) =& \sum_{k=1}^{K} \pi_k \delta_{B_k}(B) \\ \pi_k \sim& Beta(a_0/K, b_0(K-1)/K) \\ B_k \sim& H_0 \end{aligned} \tag{4.5}$$

with a valid measure as $K \to \infty$. The expression $\delta_{B_k}(B)$ equals one if $B = B_k$. Therefore, $H(B)$ represents a vector of K probabilities, with each associated with a respective atom B_k. In the limit $K \to \infty$, $H(B)$ corresponds to an infinite-dimensional vector of probabilities, and each probability has an associated atom B_k drawn i.i.d. from H_0. The choice of a_0 and b_0 is able to impose sparse constrain to matrix B. In the case where $b_0 = 1$, the marginalized beta process is equivalent to the Indian buffet process [57]. Usually, directly sampling H from the infinite beta process is difficult, but a marginalized approach IBP is often used for sampling from Beta process [14].

Originally, Beta process is defined as a prior over binary matrices with an infinite number of columns. Assuming the infinite number of columns of BP is K, $K \to \infty$. A finite approximation to the Beta process H can be made by simply setting K to a large, but finite number. This derives a Beta-Bernoulli process (BeP) approximation for the Beta process as [5, 109]:

$$\begin{aligned} \mathbf{z_i} &\sim \prod_{k=1}^{K} \text{Bernoulli}(\pi_k) \\ \pi_k &\sim \text{Beta}(a_0/K, b_0(K-1)/K), \end{aligned} \tag{4.6}$$

where π_k is the kth component of π, and a_0 and b_0 are model parameters.

In our NMF model, through the choice of a_0 and b_0 we can impose our prior belief about the sparseness of the NMF factors. Specifically, by marginalizing out the vector $\pi_1, ..., \pi_K$, it can be shown that the value of $\{z_k\}_{k=1,K}$ equal to one is distributed as Binomial$(K, a_0/(a_0 + b_0(K - 1)))$, and the expected number of ones is $a_0K/[a_0 + b_0(K - 1)]$ [24].

4.3 Hierarchical Model

Depending on the nonnegative property of BeP and exponential distribution, we can derive a simple nonparametric Bayesian NMF model for $\mathbf{X} = \mathbf{AB} + e$ by giving an exponential prior to matrix $\mathbf{A}$ and BeP prior to $\mathbf{B}$ or conversely, due to the symmetry of matrix $\mathbf{A}$ and $\mathbf{B}$. Here, $\mathbf{X} \in R^{M\times N}$, $\mathbf{A} \in R^{M\times k}$ and $\mathbf{B} \in R^{k\times N}$. When applying the BeP prior to $\mathbf{B}$, $\mathbf{B}$ will be a binary matrix. However, this is highly restricted for the factors of the NMF problem. To address this, we draw weights from another exponential distribution $\mathbf{w} \sim \prod_{k,j} \varepsilon(w_{k,j}; \beta_0)$, where β_0 is a hyperparameter of the exponential distribution. The coefficient vectors are now $\mathbf{b_i} = \mathbf{z_i} \odot \mathbf{w_i}$, and $\mathbf{x_i} = \mathbf{Ab_i}+e_i$, where $\odot$ represents Hadamard (elements-wise) multiplication of two vectors. The hierarchical form of the model can be expressed as follows:

$$
\begin{aligned}
\mathbf{x_i} =& \mathbf{A}\mathbf{b_i} + e_i \\
\mathbf{b_i} =& \mathbf{z_i} \odot \mathbf{w_i} \\
\mathbf{z_i} \sim& \prod_{k=1}^{K} \text{Bernoulli}(\pi_k) \\
\pi_\mathbf{k} \sim& \text{Beta}(a_0/K, b_0(K-1)/K) \\
A \sim& \prod_{i,k} \varepsilon(A_{i,k}; \alpha_0) \\
\mathbf{w_i} \sim& \prod_{k} \varepsilon(w_{i,k}; \beta_0) \\
e_i \sim& \mathcal{N}(0, \sigma^2) \\
\sigma^2 \sim& \mathcal{G}^{-1}(\kappa, \theta),
\end{aligned}
\tag{4.7}
$$

where $\varepsilon(x;\lambda) = \lambda \text{exp}(-\lambda x)u(x)$ is the exponential density, and $\mathcal{G}^{-1}$ is a non-informative inverse gamma prior for the noise variance.

The joint distribution of the model can be expressed as

$$
\begin{aligned}
p(\mathbf{X}, \mathbf{A}, \mathbf{w}, \mathbf{Z}, \pi, \sigma^2) = & \prod_{i,j} \mathcal{N}(\mathbf{X}_{i,j}; (\mathbf{A}(\mathbf{Z} \odot \mathbf{W}))_{i,j}, \sigma^2) \\
& \prod_{i,j} \varepsilon(A_{i,j}; \alpha_0) \prod_{i,j} \varepsilon(w_{i,j}; \beta_0) \\
& \prod_{j} \text{Beta}(\pi_\mathbf{j}; a_0, b_0) \\
& \prod_{i=1}^{N} \prod_{k=1}^{K} \text{Bernoulli}(z_{ik}; \pi_\mathbf{k}) \\
& \mathcal{G}^{-1}(\sigma^2; k, \theta)
\end{aligned}
\tag{4.8}
$$

In this chapter, we assume the sparse representation satisfies a local constraint by a graph. Specifically, when inferring the sparse coefficient of the kth sample Z_k, we refer the coefficients of the neighborhood samples on the graph. It is very likely that the kth sample will have the similar coefficient with the coefficients of the graph neighbors.

Let the nth binary sparse coefficient of the sample $\mathbf{x}_i$ be Z_n, which is a K dimensional vector. K is the rank of the dictionary. We introduce a graph to define the neighbor of Z_n. The neighbor

of Z_n is defined as:

$$\mathbb{N}[Z_n] = \{Z_{\{i\}} : G_{i,n} > 0, \}, \tag{4.9}$$

where $G_{i,n}$ is the graph weight between data $\mathbf{x}_i$ and $\mathbf{x}_n$. Then, the model for $\mathbf{z}_i$ and π_k regularized by the graph is then modified as

$$\begin{aligned} \mathbf{z}_i &\sim \prod_{i=1}^{K} \text{Bernouli}(\pi_i, n), \quad i = 1, ..., K \\ \pi_{kn} &\sim \begin{cases} \text{Beta}(\alpha_H, \beta_H), \text{if mean}(\mathbb{N}[Z_n]) \geq \epsilon) \\ \text{Beta}(\alpha_L, \beta_L), \text{if mean}(\mathbb{N}[Z_n]) < \epsilon), \end{cases} \end{aligned} \tag{4.10}$$

where ϵ is a small positive real number.

The constraint in Eq.4.10 means that the sample $\mathbf{x}_n$ will have a high probability of non-zero coefficients if its neighbor's coefficients are non-zeros as we expected. Inversely, it will have a high probability of zero coefficients. Here, we let $\alpha_L/(\alpha_L + \beta_L) \approx 0$, and $\alpha_H/(\alpha_H + \beta_H) \approx 1$.

In the next section, we will proceed for the posterior by deriving a Markov chain Monte Carlo (MCMC) sampling method.

4.4 Gibbs Sampling Inference

Posterior inference is the central computational focus for analyzing data $\mathbf{X}$ in our nonparametric Bayesian NMF model. In this section, we will use Gibbs sampling to compute the posterior distributions.

The whole inference process is summarized in Algorithm 2.

For the model with a graph constraint, the sampling of Z_{mn} and π_{mn} for $m = 1, ..., M, n = 1, ..., N$ can be modified as

$$\pi_{mn} \sim \begin{cases} \text{Beta}(\alpha_H + Z_{mn}, \beta_H + 1 - Z_{mn}), \text{ if mean}(\mathbb{N}[z_n]) \geq \epsilon), \\ \text{Beta}(\alpha_L + Z_{mn}, \beta_L + 1 - Z_{mn}), \text{ if mean}(\mathbb{N}[z_n]) < \epsilon). \end{cases} \tag{4.16}$$

Algorithm 2 Proposed Nonparametric Bayesian non-negative matrix factorization Inference.

Input:

- Input: Matrix $\mathbf{X}$.

Graph construction: G.

Initialize: Matrices and parameters: $\mathbf{A}$, $\mathbf{Z}$, $\mathbf{W}$, K, σ^2, a_0, b_0, α_0, β_0, κ, θ .

for MCMC iteration times **do**

Sample A:

$$p(A_{ik}|-) \propto \mathcal{N}(x; \mu_{A_{ik}}, \sigma_{A_{ik}^2})\varepsilon(x; \lambda), \tag{4.11}$$

where, $\mu_{A_{ik}} = \frac{\sum_j (\mathbf{x}_{ij} - \sum_{k' \neq k} A_{ik'} B_{k'j}) B_{kj}}{\sum_j B_{kj}^2}$, and $\sigma_{A_{ik}}^2 = \frac{\sigma^2}{\sum_j B_{kj}^2}$.

Sample Z**:**

$$Z_{ik} \sim \text{Bernoulli}(\frac{p_1}{p_0 + p_1}), \tag{4.12}$$

where $p_1 = \pi_k \text{exp}(-\frac{1}{2}\nu(W_{ik}^2 \mathbf{a}_k^T \mathbf{a}_k - 2W_{ik}\mathbf{a}_k^T)\widetilde{\mathbf{x}}_i^{-k})$, $\widetilde{\mathbf{x}}_i^{-k} = \mathbf{y}_i - A(\mathbf{z}_i \odot \mathbf{w}_i) + \mathbf{a}_k(Z_{ik} \odot W_{ik})$ and $p_0 = 1 - \pi_k$.

Sample π**:**

$$\pi_k \sim \text{Beta}(\frac{a_0}{K} + \sum_{i=1}^{N} Z_{ik}, \frac{b_0(K-1)}{K} + N - \sum_{i=1}^{N}(Z_{ik})). \tag{4.13}$$

Sample W:

$$p(W_{kj}|-) \propto \mathcal{N}(x; \mu_{B_{kj}}, \sigma_{B_{kj}^2})\varepsilon(x; \beta_0), \tag{4.14}$$

where
$\mu_{B_{kj}} = \frac{\sum_i (X_{ij} - \sum_{k' \neq k} A_{ik'} B_{k'j}) A_{ik}}{\sum_i A_{ik}^2}$, and $\sigma_{B_{kj}}^2 = \frac{\sigma^2}{\sum_i B_{kj}^2}$.

Sample σ^2**:**

$$p(\sigma^2|\mathbf{X}, \mathbf{A}, \mathbf{B}) \propto \quad \mathcal{G}^{-1}(\sigma^2; \kappa_{\sigma^2}, \theta_{\sigma^2}), \tag{4.15}$$

where $\kappa_{\sigma^2} = \frac{i \times j}{2} + k, \quad \theta_{\sigma^2} = \frac{1}{2}\sum_{i,j}(\mathbf{X} - \mathbf{AB})_{i,j}^2 + \theta$.
end

Output:

- Nonnegative factors $\mathbf{A}, \mathbf{B}$, where $\mathbf{B_i} = \mathbf{z_i} \odot \mathbf{w_i}$.

To reduce the model complexity and increase robustness, in our experiments, the priors for $\mathbf{A}$ and $\mathbf{B}$ are chosen to be $\alpha_0 = 10^{-6}$ and $\beta_0 = 10^{-6}$ to match the amplitude of the data.

More details of the inference are attached in the Appendix B.

4.5 Experimental Results

To illustrate the performance of the proposed method, we presents some experimental factorization results on two data sets suggested in [96, 110]: ORL face image database [90] and Digit image database MNIST. We compute the residuals and sparseness measures of different methods.

The residual is computed as [96]:

$$\|\mathbf{X} - \mathbf{AB}\|_F / \|\mathbf{X}\|_F. \tag{4.17}$$

The sparseness is computed by [95]

$$\text{sparseness}(\mathbf{x}) = \frac{\sqrt{k} - (\|\mathbf{x}\|_1)/\|\mathbf{x}\|_2}{\sqrt{k} - 1}, \tag{4.18}$$

where k is the dimensionality of $\mathbf{x}$.

4.5.1 A Simple Image Example

Figure 4.1 illustrates the generated Markov chain of the proposed NMF method for a face image factorization from the ORL dataset. Figure 4.2 shows the original image and the reconstructed image. The hyper-parameters a_0, b_0, α_0 and β_0 are typically set to be 10^{-6}. We initialize $K = 25$ and run 200 iterations of the MCMC. The chain starts to settle after about 20 iterations. From Fig. 1, we can see that the average sparseness of the vectors in the factor matrix $\mathbf{B}$ is *Sparseness =0.4490*. The residual of the reconstruction image is *Residual =0.0460*.

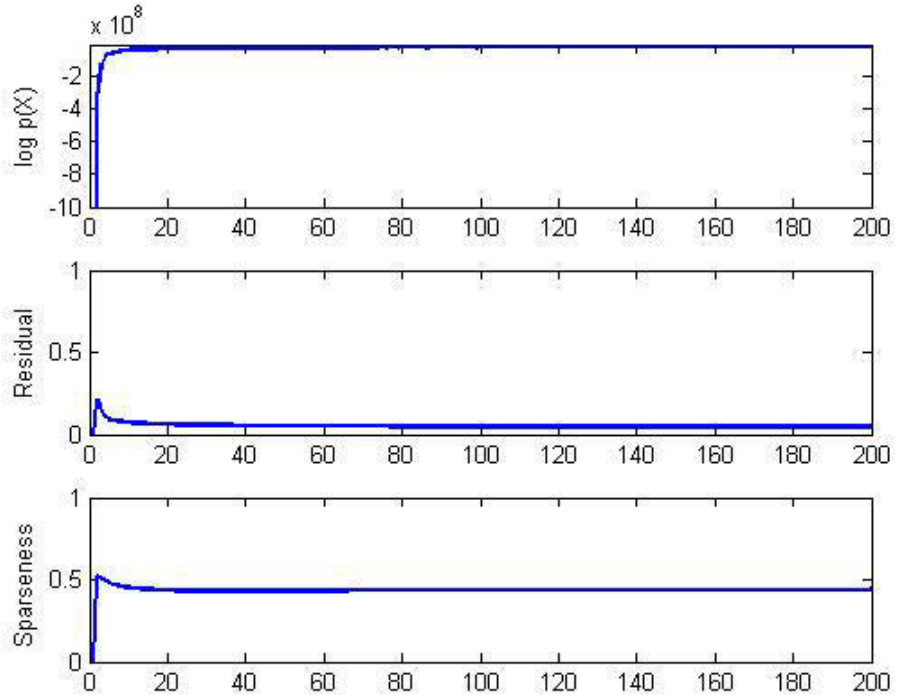
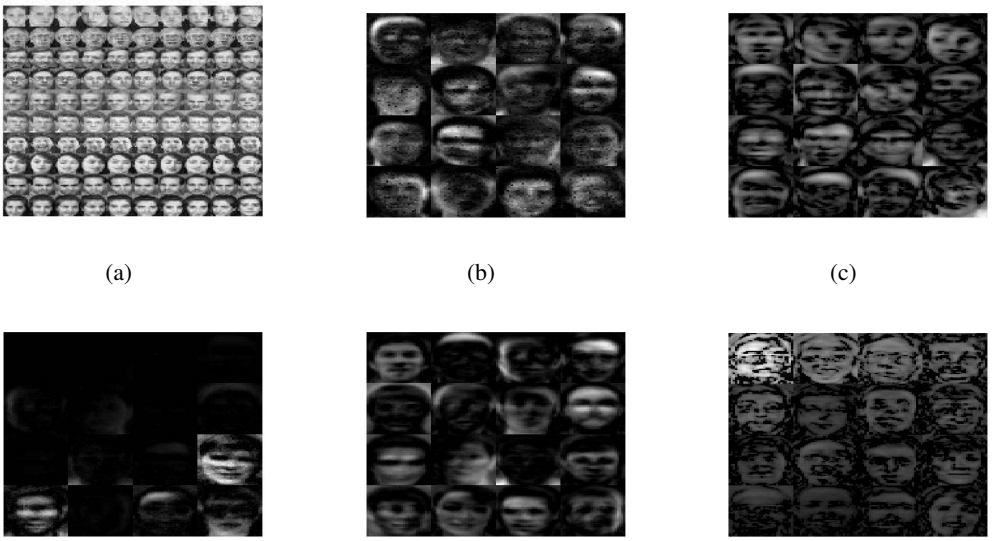


FIGURE 4.1: Generated Markov chains from Algorithm 2. Top: log posterior probability of $\mathbf{X}$; Middle: the residual computed by Equation 4.17; Bottom: average sparseness of factor $\mathbf{B}$.

(a) (b)

FIGURE 4.2: A facial image NMF by Algorithm 2. (a) original image, and (b) reconstructed image by the NMF factors from the proposed method.



(a) (b) (c)

(d) (e) (f)

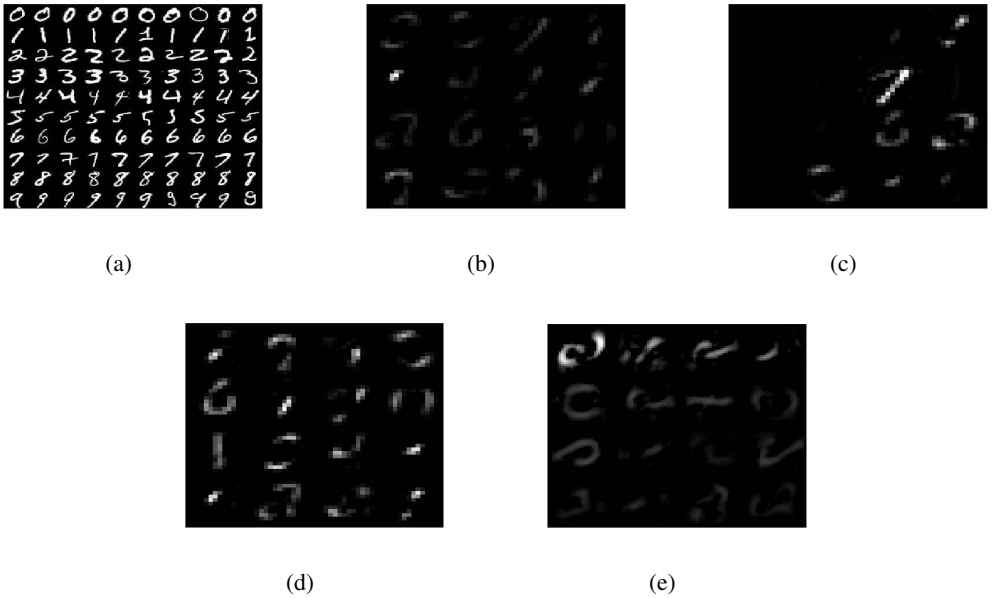
FIGURE 4.3: NMF on the ORL face image dataset. (a) ORL face image dataset; Factors computed by (b) Lee's method, (c) Lin's method, (d) Schmidt's method, (e) Hoyer's method, and (f) proposed method.

Method	Lee's	Lin's	Schmidt's	Hoyer's	Ours
Sparseness	0.264	0.339	0.338	0.342	**0.430**
Residual	0.183	0.160	0.165	0.160	**0.100**

TABLE 4.1: Results based on the ORL face database.

4.5.2 Face Dataset Example

Figure 4.3 presents the factors computed on facial ORL dataset. The images collected from ORL dataset are human faces of 40 people with different poses. Each image is represented as a vector of pixel gray values. Without loss of generality, the rank of the factors is set to be $K = 16$. We compare the results with those based on Lee's (multiplicative update) method [111], Lin's (projected gradient) method [100], Schmidt's (Bayesian) method [104] and Hoyer's (sparse constrained) method [95]. We plot the factor vectors in the form of gray-scale image with the same size of the input images in Figure 4.3. The sparseness and residual results are presented in Table 4.1. Our method achieves higher sparseness with lower residual on the dataset compared with the other methods.



(a) (b) (c)

(d) (e)

FIGURE 4.4: NMF on MNIST image dataset. (a) MNIST image dataset; Factors computed by (b) Lin's method, (c) Schmidt's method (d), Hoyer's method, and (e) proposed method.

Method	Lin's	Schmidt's	Hoyer's	Ours
Sparseness	0.727	0.726	0.753	**0.765**
Residual	0.430	0.464	**0.428**	0.430

TABLE 4.2: Results based on the MNIST digit image database.

4.5.3 Digit Dataset Example

The MNIST dataset is a dataset of handwritten digits, comprising 60 000 training examples and 10 000 test examples, which is available at `http://yann.lecun.com/exdb/mnist/index.html`.

We randomly select 400 images in our experiment. Without loss of generality, the rank of the factors is set to be $K = 16$. The digit images in the MNIST dataset are sparse themselves. Lee's (multiplicative update) method can hardly give proper representations of the factors. Figure 4.4 shows the factors computed by the other methods. Table 4.2 presents the sparseness and residual results. From Figure 4.4, we can see that, the Bayesian nonparametric prior sparse constrain gives competitive parts presentation.

4.5.4 Performance Analysis

Compared with other NMF algorithms, the proposed method can give a good sparse NMF for the given data due to the following reasons. Firstly, BeP as an implementation of Beta process is a flexible nonparametric prior with a sparse property. As we can see, the factor $\mathbf{Z}$ is a binary matrix. As the expectation $\pi \sim \beta(\alpha_0, \beta_0)$ is $\mathbb{E}[\pi_k] = \alpha_0/(\alpha_0 + \beta_0)$, we can easily impose the sparseness constraint on $\mathbf{Z}$ by let $\alpha_0/(\alpha_0 + \beta_0) \approx 0$, as it is probable that π is close to zero.

Secondly, the graph regularization give us a local constraint that similar data should have similar sparse coefficients. This constraint is natural for most image databases. In this way, it is able to better reflect the data structures as being presented in the experimental results as lower residuals on the same matrix factors rank.

4.6 Discussion and Conclusion

In this chapter, a Bayesian nonparametric framework for NMF problem has been proposed. Sparseness study is an important topic for NMF. We have utilised the BeP property, in such a way to learn the factorized matrix, as well as giving the matrix a sparseness treatment. In order to maintain their positivity, exponential distribution was chosen in this work, primarily due to its exclusive support in the positive domain, as well as its conjugacy property with Gaussian likelihood. We have given a graph regularization in our model that constrains the similar training samples having similar sparse coefficients and share similar dictionary atoms. In this way, the structure of data can be better represented.

An efficient Gibbs sampler has been developed to infer all the parameters (and latent variables) of the NMF factors. Our work has achieved its superiority which has been demonstrated in the experimental results. In our future work, we will experiment other positive-defined and conjugate prior, when non-Gaussian likelihood is present in an application, for example, the Inverse-Gamma-Weibull pair.

Like any nonparametric Bayesian method, their limitations are often associated with the speed of inference through the use of Gibbs sampling. Our future work is to focus on developing a parallel sampling strategy by introducing auxiliary variables.

Chapter 5

A Graph-regularized Nonparametric Bayesian method for Face Recognition

5.1 Introduction

Face recognition (FR) is one of the most important and challenging research topics in computer vision, machine learning and biometrics. One popular method for this problem is sparse representation-based classification (SRC) or sparse coding recognition (SCR).

Sparse representation methods have been well applied in computer vision and machine learning areas, such as feature representation and selection, dictionary learning, data clustering, semi-supervised learning, subspace learning and so on. Specially, it is well applied in the problem of face recognition [112–115]. It can be seen as a further generalization of nearest neighbor (NN) or nearest subspace (NS) methods which are based on the best representation of the test image by the training samples [116–118]. The problem solved by the sparse representation is to search for the most compact representation of a signal in terms of linear combination of atoms in an overcomplete dictionary [119]. Let A be a matrix of training sample vectors, y is a testing sample vector. The basic idea of SR based classification can be described as [115]:

1. Normalize the columns of A to have unit l_2-norm;

2. Solve the l_1-minimization problem

$$\hat{\alpha}_1 = \arg\min_{\alpha} \|\alpha\|_1 \qquad s.t. \|A\alpha - \mathbf{y}\|_2 \leq \varepsilon; \tag{5.1}$$

3. Compute the residuals;

4. Classification by finding the minimum of residuals.

The sparse representation based classification method can somewhat overcome the challenging issues from illumination changes, random pixel corruption, large block occlusion or disguise, and so on [120]. More and more researchers have been applying the sparse representation theory to the fields of computer vision and pattern recognition, especially in image classification.

Wright *et al.* proposed the classical method for employing sparse representation to perform robust face recognition [115]. In their work, the training face images were used as a dictionary of representative samples, and an input testing image was coded as a sparse linear combination of these sample images via l_1-norm minimization. Many revised SRC methods have been proposed to improve the performance.

For the FR problem, usually, it is necessary to reduce the face image dimension for improving robustness. Different methods like PCA, Randomfaces, Fisherfaces are employed in SRC based FR for this purpose [116, 121]. Zhang *et al.* proposed a dimensionality reduction of images under the framework of SRC [116]. A well trained discriminative projection matrix was learned to reduce the dimension of original images. It got better performance on some databases than the original SRC. Similar work proposed in [122] employed a random matrix to exploit the presumed sparsity in the FR problem. Specifically, generated a random matrix $\Phi \in \mathbb{R}^{d \times m}$ where $d \ll m$ and identified the vector α which minimises the following l_1 problem:

$$\hat{\alpha}_1 = \arg\min_{\alpha \in \mathbb{R}^{\kappa}} \|\alpha\|_1 \quad s.t. \|\mathbf{\Phi x} - \mathbf{\Phi A}\alpha\|_2 \leq \varepsilon. \tag{5.2}$$

Ma *et al.* proposed a discriminative low-rank dictionary learning algorithm for sparse representation [123]. They assumed the data from the same pattern were linearly correlated, and the dictionary should be approximately low-rank. An objective function with sparse coefficients, class

discrimination and rank minimization was proposed and optimized during dictionary learning.

Shortcomings

In the SRC based image classification work, it is widely believed that the sparsity constraint on coding coefficients plays a key role in its success. However, the sparsity assumption which underpins much of this work is not always supported by the training data [124, 125]. The lack of sparsity in the data means that compressive sensing approach cannot be guaranteed to recover the exact signal, and therefore that sparse approximations may not deliver the robustness or performance desired.

An important fact in FR is that all the face images are somewhat similar, while some subjects may have very similar face images. However, most present SCR based FR methods find the sparsest representation of each sample individually, lacking global or local constraints. For example, if we directly use training data, X_i, to present the testing sample y, the representation error can be big, even when y is from class i. This drawback can greatly reduce the performance when the data is grossly corrupted. Consequently, the classification will be unstable if the error e_i or the sparsity $\|\alpha_i\|_p$ is used, or both of them are used for experiments.

To solve this problem, some researchers consider other constraints to exploit the presumed sparsity of the FR problem. Authors in [124–126] relaxed the l_1 minimization by l_2 which collaboratively represented the query sample to improve the performance [124]. Wang *et al.* proposed a manifold regularized local sparse representation model for FR problem [127]. The key idea behind this method is that all coding vectors in sparse representation should be group sparse, which means holding the two properties of both individual sparsity and local similarity. As a consequence, the face recognition rate can be considerably improved.

5.2 Proposed Method

Motivated by the progress of Bayesian nonparametric model and graph using in computer vision and machine learning applications [106–108, 128], in this chapter, we propose a novel algorithm, called graph-regularized nonparametric Bayesian method for FR. We assume that the similar

training samples in the graph should have similar sparse coefficients and they share the similar dictionary atoms. In this way, our method minimizes the reconstructed error and improves the discriminative ability of the training samples for the sparse representation. It brings more accurate representations of the test samples, especially, corrupted samples like occlusion or disguise. The experimental results on several benchmark face image databases manifest the effectiveness and robustness of our proposed method.

5.2.1 Graph Construction

We assume that the sparse representation of the training data satisfies a local constraint by a graph. Specifically, when inferring the sparse coefficient of the kth sample $\mathbf{x}_k$, we refer to the coefficients of the neighborhood samples on the graph. It is very likely that the kth sample will have the similar coefficient to the coefficients of the graph neighbors.

The edge between $\mathbf{x}_i$ and $\mathbf{x}_j$ is decided by the K-nearest-neighbor (Knn) method. The similarity between two data can be decided by different measures. The different similarity measures are suitable for different situations. While for image data, the heat kernel is a popular choice [106]. Since W_{ij} in our model is only for measuring the closeness, we do not treat the different weighting schemes separately. The graph weight between x_i and x_j is then computed by heat kernel shown in Eq. 4.4.

We use the Euclidean distance to estimate the dissimilarity distance between the data vectors. For example, the distance between data x_i and x_j is defined as: $d_{ij} = \|\mathbf{x_i} - \mathbf{x_j}\|_2$. Then, we get a symmetric graph $G(V, E)$. In order to improve the robustness for noise, we use PCA features when measuring the dissimilarity.

5.2.2 Bayesian Nonparametric Model

Given observed training data set $X = [\mathbf{x}_1, ..., \mathbf{x}_N] \in \mathbb{R}^{M \times N}$, in the training stage, our objective is to learn the sparse representation by a dictionary and the corresponding sparse coefficients. The Bayesian nonparametric model is

$$\mathbf{X} = \mathbf{D}(\mathbf{Z} \odot \mathbf{W}) + e. \tag{5.3}$$

In Eq. 5.3, e is a noise term drawn independently and identically from a Gaussian distribution, $\odot$ denotes the Hadamard product, each column $\mathbf{x}_i$ in $\mathbf{X}$ is a vector expanded by a training image, $\mathbf{D} \in \mathbb{R}^{M\times K}$ is the dictionary, $Z \in \mathbb{R}^{K\times N}$ is sparse coefficients and $\mathbf{W} \in \mathbb{R}^{K\times N}$ is weight matrix, where the integer K defines the rank of the dictionary and the sparse coefficient matrix. This model has been well used in image completion and image denoising [14, 16]. It suggests that each column of $\mathbf{X}$ can be viewed as a weighted sum of the dictionary elements (columns) in $\mathbf{D}$, and the K can be seen as the size of the dictionary. The weights globally determine all the dictionary elements that are active to the training data, and also determine the importance of the selected dictionary elements for representation of column of $\mathbf{X}$. Below, we explain the model in detail.

The columns of basic vectors $\mathbf{d}_k, k = 1, ..., K$ are assumed to be drawn from a normal distribution:

$$\mathbf{d}_k \sim N(0, \frac{1}{M}\mathbf{I}_M), k = 1, 2, ..., K, \tag{5.4}$$

where, $\mathbf{I}_M$ denotes an $M \times M$ identity matrix. To reduce the model complexity and increase robustness, we let all the data in $\mathbf{D}$ share the same precision parameter.

Matrix $\mathbf{Z}$ is a binary matrix given by a Beta-bernoulli process prior, which is modeled as:

$$\mathbf{z}_i \sim \prod_{k=1}^{K} \text{Bernoulli}(\pi_k), \tag{5.5}$$

where π_k is the kth component of π. Given parameters $\pi = \{\pi_n\}_{n=1,2,...,N}$, the entries in the binary matrix $\mathbf{Z}$ are assumed to be drawn independently. π_k is given a Beta distribution prior as,

$$\pi_k \sim \text{Beta}(a_0/K, b_0(K-1)/K), \tag{5.6}$$

where hyperparameters $\alpha_0 > 0$ and $\beta_0 > 0$. The sparseness of matrix $\mathbf{Z}$ can be controlled by α_0 and β_0. As the expectation π is $\mathbb{E}[\pi_k] = \alpha_0/(\alpha_0 + \beta_0(K-1))$, to impose the sparseness on $\mathbf{Z}$, we may let $\alpha_0/(\alpha_0 + \beta_0(K-1)) \approx 0$. Therefore, it is probable that π is close to zero.

Here, we assume that the sparse representation satisfies a local constraint by a graph. Specifically, when inferring the sparse coefficient of the kth sample $\mathbf{z}_k$, we refer to the coefficients of the

neighborhood samples on the graph. It is very likely that the kth sample will have the similar coefficient to the coefficients of the graph neighbors.

We introduce the graph to define the neighbor of Z_n. The neighbor of Z_n is defined as $\mathbb{N}[Z_n] = \{Z_{\{i\}} : G_{i,n} > 0\}$, which is similar to the previous chapter. Here, $G_{i,n}$ is the graph weight between data $\mathbf{x}_i$ and $\mathbf{x}_n$. Then, the model of $\mathbf{z}_i$ and π_k regularized by the graph is then modified:

$$\begin{aligned} \mathbf{z}_i &\sim \prod_{k=1}^{K} \text{Bernouli}(\pi_{in}), \quad k = 1, ..., K \\ \pi_{kn} &\sim \begin{cases} \text{Beta}(\alpha_H, \beta_H), \text{ if } \mathcal{Z}_k \geq \epsilon \\ \text{Beta}(\alpha_L, \beta_L), \text{ if } \mathcal{Z}_k < \epsilon, \end{cases} \end{aligned} \tag{5.7}$$

where, $\mathcal{Z}_k = \text{mean}(\mathbb{N}[Z_k])$, and ϵ is a small positive real number. We let $\alpha_L/(\alpha_L + \beta_L) \approx 0$, and $\alpha_H/(\alpha_H + \beta_H) \approx 1$,

The graph constrains that the sample image $\mathbf{x}_n$ will have a high probability of non-zero coefficients if its neighbor image sample coefficients are non-zeros. Inversely, it will have a high probability of zero sparse coefficients. This constraint means that similar face images share atoms in the dictionary by similar coefficients. And different face images own different atoms which will be beneficial to the classification of testing samples.

The columns of coefficients $\mathbf{w}_m, m = 1, ..., M$ are assumed to be drawn from a normal-gamma distribution.

$$\mathbf{w}_n \sim N(0, \frac{1}{\nu}\mathbf{I}_K), n = 1, ..., N \tag{5.8}$$

$$\nu \sim \text{Gamma}(c_0, d_0), \tag{5.9}$$

where, $\mathbf{I}_M$ denotes an $M \times M$ identity matrix. We let all the data in $\mathbf{X}$ share the same precision parameter ν, to reduce model complexity and increase robustness. Typically, we set $c_0 = d_0 = 10^{-6}$.

Finally, for the noise item, we assume the measurement i.i.d noise is drawn from a Gaussian distribution with variance hyperparameter γ from a Gamma distribution. The noise variance or precision is assumed unknown, and is learned within the model inference.

In our applications, each column of $\mathbf{X}$ corresponds to an image, and each image may have its own noise level. Mathematically, the noise is modeled as

$$\begin{aligned} e_{mn} &\overset{\text{i.i.d}}{\sim} \mathcal{N}(0,\gamma), \quad m = 1,2,...,M, \;\; n = 1,2,...,N, \\ \gamma &\sim \text{Gamma}(e_0, f_0), \end{aligned} \tag{5.10}$$

where, e_{nm} denotes the entry at row n and column m of e, e_0 and f_0 are hyper-parameters. Typically, we set hyperparameters $e_0 = f_0 = 10^{-6}$.

In the next section, we will proceed for the posterior by deriving a Markov chain Monte Carlo (MCMC) sampling method.

5.2.3 Gibbs Sampling Inference

In this section, we will use Gibbs sampling to compute the posterior distribution of different parameters.

Sample $\mathbf{d}_k$:

The columns of $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_M]$ are assumed drawn from a Gaussian distribution. According to the Bayesian formula, the posterior of $\mathbf{d}_k$ can be expressed as [109]:

$$p(\mathbf{d}_k|-) \propto \prod_{i=1}^{N} \mathcal{N}(\mathbf{y}_i; \mathbf{D}(\mathbf{w}_i \odot \mathbf{z}_i), \nu^{-1}\mathbf{I}) \mathcal{N}(\mathbf{d}_k; 0, P^{-1}\mathbf{I}_P). \tag{5.11}$$

As the conjugation of Gaussian distributions, it can be shown that $\mathbf{d}_k$ is drawn from

$$p(\mathbf{d}_k|-) \sim \mathcal{N}(\mu_{\mathbf{d_k}}, \mathbf{\Sigma_{d_k}}) \tag{5.12}$$

with the covariance $\mathbf{\Sigma_{d_k}} = (P\mathbf{I} + \nu \sum_{i=1}^{N} z_{ik}^2 w_{ik}^2)^{-1}$ and the mean $\mu_{\mathbf{d_k}} = \nu \Sigma_{d_k} \sum_{i=1}^{N} z_{ik} w_{ik} \widetilde{\mathbf{x}}_i^{-k}$, where $\widetilde{\mathbf{x}}_i^{-k} = \mathbf{y}_i - D(\mathbf{z}_i \odot \mathbf{w}_i) + \mathbf{d}_k(z_{ik} \odot w_{ik})$.

Sample $\mathbf{z_k}$:

Let $\mathbf{z_{k:}} = [z_{1k}, z_{2k}, ..., z_{Nk}]$, the posterior of $\mathbf{z_{k:}}$

$$p(z_{ik}|-) \propto \mathcal{N}(\mathbf{y}_i; D(\mathbf{w}_i \odot \mathbf{z}_i), \nu^{-1}\mathbf{I})\text{Bernoulli}(z_{ik}; \pi_k). \tag{5.13}$$

The posterior probability that $z_{ik} = 1$ is proportional to

$$p_1 = \pi_k \exp(-\frac{1}{2}\nu(w_{ik}^2 \mathbf{d}_k^T \mathbf{d}_k - 2w_{ik}\mathbf{d}_k^T)\widetilde{\mathbf{x}}_i^{-k}) \tag{5.14}$$

and the posterior probability that $z_{ik} = 0$ is proportional to

$$p_0 = 1 - \pi_k. \tag{5.15}$$

Therefore, z_{ik} can be drawn from a bernoulli distribution as

$$z_{ik} \sim \text{Bernoulli}(\frac{p_1}{p_0 + p_1}). \tag{5.16}$$

Sample π:

$$\pi_k|- \propto \text{Beta}(\pi_k; a_0, b_0) \prod_{i=1}^{N} \text{Bernoulli}(z_{ki}; \pi_k). \tag{5.17}$$

It can be shown that π_k can be drawn from a Beta distribution as

$$\pi_k \sim \text{Beta}(\frac{a_0}{K} + \sum_{i=1}^{N} z_{ki}, \frac{b_0(K-1)}{K} + N - \sum_{i=1}^{N}(z_{ki})). \tag{5.18}$$

For the model with a graph constraint, the sampling of Z_{mn} and π_{mn} for $m = 1, ..., M, n = 1, ..., N$ can be modified as follows:

$$\pi_{ik} \sim \begin{cases} \text{Beta}(\alpha_H + z_{ik}, \beta_H + 1 - z_{ik}), \text{ if } \mathcal{Z}_k \geq \epsilon \\ \text{Beta}(\alpha_L + z_{ik}, \beta_L + 1 - z_{ik}), \text{ if } \mathcal{Z}_k < \epsilon. \end{cases} \tag{5.19}$$

Here, we let $\alpha_L/(\alpha_L + \beta_L) \approx 0$, $\alpha_H/(\alpha_H + \beta_H) \approx 1$, and $\mathcal{Z}_k$ and ϵ are defined as Eq. 5.7.

The Eq. 5.19 constrains the sample k has high probability of non-zero coefficients, if its neighbors coefficients are non-zeros. Inversely, it will have low probability of non-zero coefficients. In this way, the samples in the same training class will contain similar sparse coefficients which is we expected.

Sample $\mathbf{w}_k$:

$$\begin{aligned} \mathbf{w}_k &\sim \mathcal{N}(0, \nu^{-1}\mathbf{I_N}) \\ p(w_{ik}|-) &\sim \mathcal{N}(\mu_{w_{ik}}, \Sigma_{w_{ik}}), \end{aligned} \tag{5.20}$$

where $\Sigma_{w_{ik}} = (\nu + \gamma z_{ik}^2 \mathbf{d}_k^T \mathbf{d}_k)^{-1}$ and $\mu_{w_{ik}} = \gamma z_{ik} \mathbf{d}_k \widetilde{\mathbf{x}}_i^{-k}$.

Sample ν:

$$p(\nu|-) \propto \text{Gamma}(\nu; c_0, d_0) \prod_{i=1}^{N} \mathcal{N}(\mathbf{w}_i; 0, \nu^{-1}\mathbf{I}_K) \tag{5.21}$$

It can be shown that ν can be drawn from a Gamma distribution as

$$p(\nu|-) \sim \text{Gamma}(c_0 + \frac{1}{2}KN, d_0 + \frac{1}{2}\sum_{i=1}^{N} \mathbf{w}_i^T \mathbf{w}_i). \tag{5.22}$$

Sample γ:

$$p(\gamma|-) \sim \text{Gamma}(\gamma; e_0, f_0) \prod_{i=1}^{N} \mathcal{N}(\mathbf{x}_i; D(\mathbf{z}_i \odot \mathbf{w}_i), \gamma^{-1}\mathbf{I}_P). \tag{5.23}$$

It can be shown that γ can be drawn from a Gamma distribution as:

$$p(\gamma|-) \sim \text{Gamma}(e_0 + \frac{1}{2}NK, f_0 + \frac{1}{2}\sum_{i=1}^{N} \|\widetilde{X}_i^{-k}\|_F^2), \tag{5.24}$$

where $\widetilde{\mathbf{x}}_i^{-k} = \mathbf{y}_i - D(\mathbf{z}_i \odot \mathbf{w}_i) + \mathbf{d}_k(z_{ik}) \odot w_{ik}$.

The whole inference process is summarized in Algorithm 3.

Algorithm 3 Inference of the proposed method

Input:

- Training dataset $X_1, ..., X_N$.

Constructing graph: $G(V, E)$ by Knn and Eq. 4.4.
Initialize Hyperparameters: $a_0, b_0, c_0, d_0, e_0, f_0$.
Repeat: In each sample iteration,

1. Sample $\mathbf{D}$ by Eq. 5.12;
2. Sample $\mathbf{Z}$ by Eq. 5.16;
3. Sample π by Eq. 5.19;
4. Sample $\mathbf{W}$ by Eq. 5.20;
5. Sample ν by Eq. 5.22;
6. Sample γ by Eq. 5.24.

Output:

- Matrices $\mathbf{D}, \mathbf{B}$, where $\mathbf{D_i} = \mathbf{z_i} \odot \mathbf{w_i}$.

The testing stage is summarized in Algorithm 4.

Algorithm 4 Testing stage of the proposed method

Input: Testing images $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, ...\}$
Process:

- Compute the projection matrix by Eq. 5.25.

Repeat: for each testing image sample $\mathbf{y}_i$

- Solve the l_1-minimization problem of Eq. 5.26;
- Compute the residuals by Eq. 5.27;
- Find the recognition label by the minimum of residuals.

Output:

- Recognition error.

So far, we have learned the dictionary D and sparse coefficients $B = Z \odot W$ from the training samples by the graph-regularized nonparametric Bayesian method. In the testing stage, firstly, we calculate the projection matrix P by

$$\mathbf{P} = (D^T D + \lambda \mathbf{I}_D) D^T \tag{5.25}$$

where λ is a fixed regular parameter, and $\mathbf{I}_D$ is a unit matrix. Typically, we set $\lambda = 0.001$ in our experiments. For each testing sample $\mathbf{y}_i$, we solve the l_1-minimization problem as the traditional SCR algorithm by:

$$\hat{\alpha} = \arg\min_{\alpha} \|\alpha\|_1 \quad s.t. \ \|\mathbf{PA}\alpha - \mathbf{y}_i\|_2 \leq \varepsilon. \tag{5.26}$$

The minimum residual error label can be found by

$$\arg\min_{\mathbf{j}} \|\mathbf{y}_i - (\mathbf{PA})_j \hat{\alpha}\|_2. \tag{5.27}$$

Finally, the label j is the recognized label.

5.3 Experimental Results

To illustrate the performance of the proposed method, we evaluate our algorithm on different popular databases including ORL face database [90] and AR face database [129].

We compare the proposed method with its competing methods, including a simple and fast representation-based face recognition (SFR) (which has better performance than nearest neighbor (NN) method)[128], sparse representation coding (SRC) [115], collaborative representation based classification (CRC) and robust collaborative representation based classification (RCRC) [124, 126].

5.3.1 ORL Face Database

The images collected from ORL database are human faces of 40 people with different poses. Each image is resized to 86×84 and represented as a vector of pixel gray values. We randomly select $n(n = 1, 2, ..., 6)$ images in each class for training. The rest images are used for testing. For fair comparison, we compute the average classification error rate for every method.

Fig. 5.1 shows an example of the constructed graph in our method. It can be seen that the heat kernel graph construction can well reflect the local relationship between the training samples.

(a) (b)

FIGURE 5.1: Graph constructed by the training samples. (a) The true graph constructed by the training labels, and (b) the graph constructed by the heat kernel. Five samples in each class are used here for graph construction in this example.

Specifically, the samples with the same label are more likely to own nonzero graph weights for the links between each other.

Fig. 5.2 shows the results of different methods with different training samples in each class from 1 to 6. We can see that the method CRC and our proposed method give better results when the training sample number is small in each class. However, when the training sample number increases, the performance of CRC is not as good as the proposed method. This is because that the graph-regularized method can obtain more similar sparse coefficients and share more the similar dictionary atoms when the training samples are increasing, which is beneficial to the recognition results.

From the experiments, we can also see that more training samples in each class usually result in lower error classification rates.

5.3.2 ORL Face Database with Occlusion

One important advantage of sparse representation (or coding) based FR methods is their ability to deal with occlusion. In this experiment, we simulate various levels of contiguous occlusion from 5% to 30%, by replacing a random located square block of each test image with an unrelated image as shown in Fig. 5.3. The location of occlusion is randomly chosen from each image. The block occlusion of a certain size is located on the random position which is unknown to the FR algorithm. This makes the methods that select fixed facial features or blocks of the image

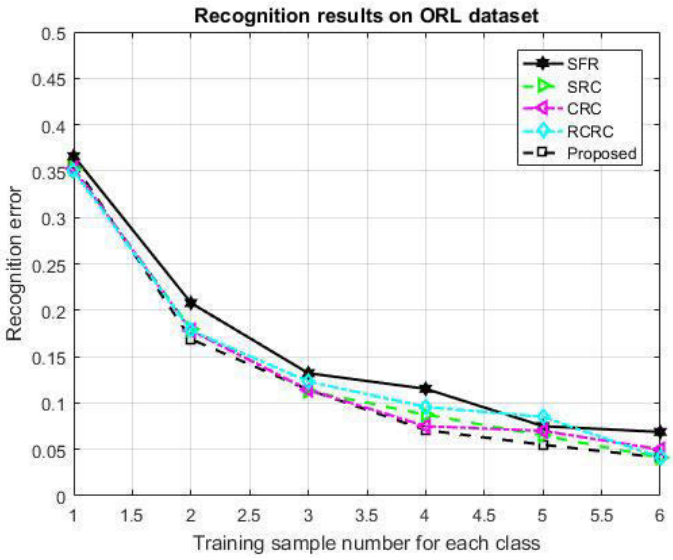


FIGURE 5.2: Recognition results on ORL database.

are less likely to succeed due to the unpredictable location of the occlusion [115]. One example of testing samples with block occlusion is shown in Fig. 5.3. The occlusions presented here have 20% blocks in size.

For the experiment with random occlusion, we analyze the relationship between the occlusion ration and the performance of different methods. We use the same experiment setting as described in the first experiment. The number of training samples in each class is fixed to 5. The results of different methods for the ORL database with random occlusions are represented in Fig. 5.4. As sparse representation methods have the property to deal with the occlusions, it is not hard to understand that SRC method, CRC method, RCRC method and the proposed method have better results than SFR method. The graph regularized nonparametric Bayesian model improves the robustness of SRC to the noise, therefore, the proposed method again outperforms the other methods.

5.3.3 AR Face Database

Original AR face database contains over 4,000 color images corresponding to 126 people's faces (70 men and 56 women). Images feature frontal view faces with different facial expressions, illumination conditions, and occlusions.

FIGURE 5.3: ORL face images with random occlusions.

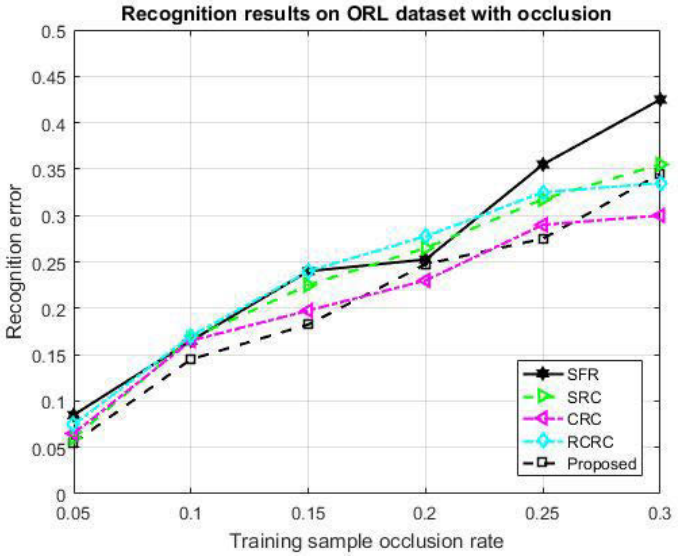


FIGURE 5.4: Recognition results on ORL database with occlusion.

As suggested in [126], we use a subset (with only illumination and expression changes) that contains 50 male subjects and 50 female subjects chosen from the AR dataset in our experiments for training. The images are cropped to 60×43. The other subset is for testing. We also vary the number of training samples of each class from 2 to 6 to study the performance of different methods with different number of training samples in every class. We estimate the recognition errors by different methods, including SFR, SRC, CRC, and RCRC. As shown in Fig. 5.5, generally speaking, the proposed method and CRC method achieve similar results which are better than others.

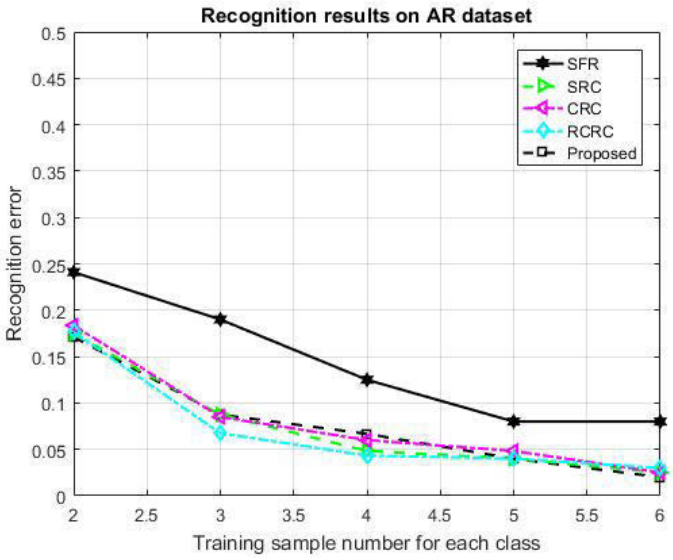


FIGURE 5.5: Recognition results on AR database.

FIGURE 5.6: Example images in AR database.

5.3.4 AR Face Database with Real Disguise

To have a more comprehensive observation of these methods robustness to disguise, we experiment on real disguise images. Fig. 5.6 presents some samples from AR database.

We use a disguise subset from the AR database for testing. The training set is same as the above experiment. As shown in Fig. 5.7, the performance of our approach can be improved more rapidly than other methods with the increasing number of training samples per individual. That is because our method can capture similar sparse coefficients and share the similar dictionary

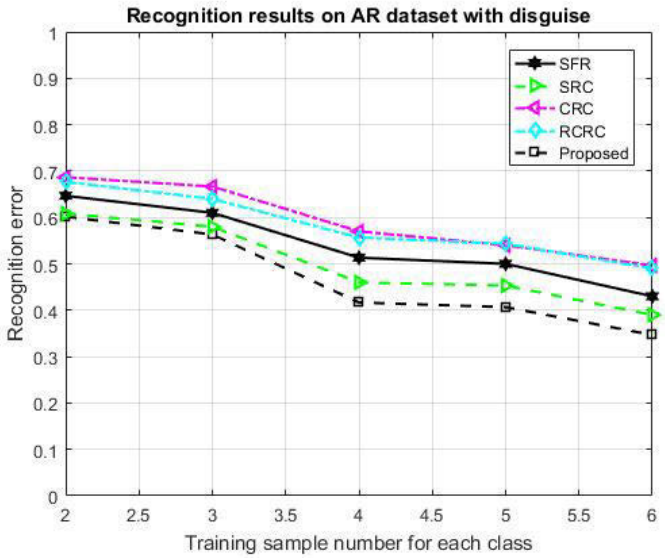


FIGURE 5.7: Recognition results on AR database with real disguise.

atoms. When each individual contains more images, the recognition error will be lower. Real disguise recognition is a very challenging topic for FR problem. This can be reflected from the performance in the experiment. Although the proposed method present better performance than others, there still are some improvements for the future work.

5.4 Conclusion

Sparse representation-based classification (SRC) has caught attention of many researchers as one of the most popular face recognition techniques. In this chapter, we presented a graph-regularized Bayesian nonparametric method based SRC for face recognition problem. We have utilised a nearest neighbor graph and Beta-Bernoulli process to get an appropriate dictionary with sparse coefficients for image recognition. The dictionary regularized by a graph has the property that the similar training samples have similar sparse coefficients and similar dictionary atoms. In this way, the testing samples are able to get more accurate representations from the training dictionary and better performance for FR, especially, when the testing samples are occluded or disguised.

We have applied the algorithm to different face image databases to demonstrate the effectiveness of our method. The experimental results have confirmed that our proposed method is robust and

effective. It has achieved a promising face recognition performance compared with the existing SCR based algorithms.

Chapter 6

Conclusions and Future work

6.1 Conclusions

In this thesis, we have presented our research work on Bayesian nonparametric models and related applications. Specifically, after an introduction of Bayesian nonparametric methods and some well developed applications in Chapter 1, preliminaries of Bayesian nonparametric statistics and advanced inference methods have been described in Chapter 2. In Chapter 3, 4 and 5, different Bayesian nonparametric models have been applied to three computer vision and machine learning problems: Face Hallucination, Sparse Nonnegative Matrix Factorization and Face Recognition.

In Chapter 2, basic notions on completely random measures and Lévy processes have been represented, which are preliminaries of Bayesian nonparametric methods. Also, two remarkable families of Lévy processes: Gamma process and Beta process have been introduced. We have discussed Lévy -Khintchine formula and Lévy-Itô decomposition of Lévy processes, as well as their relationship with completely random measures. The last part of this chapter presented an advanced Bayesian inference method Markov chain Monte Carlo (MCMC) and related sampling algorithms.

In Chapter 3, a novel example-based face hallucination method based on nonparametric Bayesian learning has been proposed. With the assumption that all human faces have similar local pixel structures, in the proposed method, the low-resolution (LR) face image patches are clustered by

using a nonparametric method, namely the distance dependent Chinese restaurant process (dd-CRP). Then, the centers of the clusters can be calculated. The dictionaries are learned for each subspace to map the LR patches to the HR patches. Therefore, the HR patches of the input LR face image can be efficiently generated by using the learned mapping dictionaries. The ddCRP gives a robust implementation of learning the clusters without setting the number of clusters in advance. The spatial distance constraint is employed to assist in learning the cluster centers and mapping dictionaries, such that each dictionary can better reflect the detailed information about image patches. Experimental results have shown that our method is efficient and can achieve competitive performance for face hallucination compared with the state-of-the-art methods.

In Chapter 4, a Bayesian nonparametric method for sparse nonnegative matrix factorization (NMF) problem has been proposed. Sparseness study is an important topic for NMF. We have utilised the Beta process property to learn the factorized matrices, as well as giving the matrices a sparseness treatment. In order to maintain their positivity, exponential distributions are chosen in this work, primarily due to their exclusive support in the positive domain and conjugacy property with Gaussian likelihood. The graph in our model regularizes the similar training samples to have similar sparse coefficients and share similar dictionary atoms, which will give a better representation of the training data. An efficient Gibbs sampler has been developed to infer all the parameters (and latent variables) of the NMF factors. Our work has achieved its superiority which has been demonstrated in the experimental results.

In Chapter 5, a novel graph-regularized Nonparametric Bayesian method for sparse coding recognition problem has been proposed. In order to get an appropriate dictionary with sparse coefficients for image recognition, we used a graph regularized Beta process (BP) prior and Gaussian distribution prior for the dictionary learning. BP is a nonparametric method which lends itself naturally to model sparse binary matrices with an infinite number of columns. The graph in our model regularizes training samples in a same class to have similar sparse coefficients and share similar dictionary atoms, which is beneficial to the reconstruction of the testing image. An efficient Gibbs sampler has been derived to approximate the posterior density of the proposed model. We demonstrated the effectiveness of our method on different image databases. The experimental results show that our proposed method gives competitive results.

6.2 Future Work

My long-term research goal is to develop novel statistical models and efficient computational tools for solving machine learning and computer vision problems. My short-term objectives are to advance the research in large-scale/high-dimensional problems by using Bayesian nonparametric models, the applications on recently developed graph construction by random measures, and hierarchical factor analysis for deep learning.

6.2.1 Advanced Inference on Large-Scale Data

The major problems for many MCMC algorithms are slow convergence and poor mixing [1]. For example, the Gibbs sampler converges slowly for the sparse dictionary learning as discussed in Chapter 3 and 4. The Gibbs sampler is a special case of the Metropolis-Hastings algorithm, has become one of the most popular inference methods due to its intuitive explanation and simple implementation. However, it is not necessarily the best choice in the general case, as discussed in Chapter 2, there are a few conditions that must be fulfilled for the Gibbs sampler to work. Firstly, it requires that the Markov chain induced by the sampler must be irreducible. If this does not hold, the sample space contains non-communicating sets that the Gibbs sampler is unable to reach, thus it will never be able to correctly estimate the required distribution. Secondly, even if the chain is irreducible, the chain may be near-reducible and mixing so slowly that it requires astronomical sample sizes to move around the sample space [130].

A promising approach to solve this problem is parallel MCMC, which firstly partitions the data into multiple subsets and runs independent sampling algorithms on each subset. The subset posterior draws are then aggregated via some combining rules to obtain the final approximation. For example, Chang and Fisher proposed a parallel sampling of DP mixture models using sub-cluster splits [131]; Wang *et al.* proposed an embarrassingly parallel MCMC algorithm which applied random partition trees to combine the subset posterior draws [132].

Another popular strategy for improving the statistical efficiency of Gibbs sampling is *Blocking*. Unlike Gibbs sampling which samples each variable individually given others, blocked Gibbs sampling partitions the variables into disjoint groups or blocks and then jointly samples all

[1] Mixing means the stationary distribution reached from an arbitrary position.

variables in each block given an assignment to all other variables not in the block [133]. Usually, blocking sampling algorithm combines a particular clustering method, such as using the junction tree propagation. This makes it possible to implement the general Gibbs sampler where the components consist of many variables instead of a single one. Thus, many variables are updated jointly, often resolving problems of reducibility and slow mixing [130].

Our development of Bayesian nonparametric models have been successfully applied in some applications. However, it is still necessary to apply advance inference methods for the future research, especially, on large scale data.

6.2.2 Graph on Random Measures

In this thesis, we have applied graph-regularization for sparse nonnegative matrix factorization and sparse coding recognition problems. Recently, the graph have been represented as an exchangeable discrete structure and considered as a measure on $\mathbb{R}_+$.

For example, Lloyd *et al.* considered nonparametric models for graphs and arrays. The random arrays that satisfying an exchangeability property can be represented in terms of a random function according to the Kallenberg representation theorem [134]. They obtained a flexible yet simple Bayesian nonparametric model by placing a Gaussian process prior on the parameter function and demonstrated applications of the model to network data [135]. The related work presented by Cai *et al.* defined priors on exchangeable directed graphs, they demonstrated theirs models on synthetic data [136].

Caron and Fox considered representing the graph as a measure on $\mathbb{R}_+$. They defined the exchangeability in a continuous space based on the Kallenberg representation theorem and showed that for certain choices of such exchangeable random measures underlying the graph construction their network process is sparse with power-law degree distribution. In particular, they built on the framework of completely random measures and used the theory associated with such processes to derive important network properties. Their method was able to recover graphs ranging from dense to sparse and well applied in a range of real data sets, including Facebook social circles, citation networks, world wide web networks and so on [137].

As introduced above, the recent research of graph construction based on nonparametric Bayesian statistics has been well applied for inferring the relationship models, such as the applications on social circles, web network and so on. It is considered a powerful and flexible modeling approach as providing a number of possible priors on random functions. As the graph-based methods have been extensively used in computer vision and machine learning problems like data clustering, subspace learning, semi-supervised learning and so on, we believe the research of random graph construction will provide more state-of-the-art performances for other computer vision and machine learning problems.

6.2.3 Hierarchical Factor Analysis for Deep Learning

In recent years, deep learning or multilayered models for representation of general data has received significant attention. Methods that have been considered include convolutional Restrict Boltzmann Machines (RBMs), deconvolutional networks, convolutional networks, deep belief networks (DBNs), hierarchies of sparse autoencoders, and so on. Deeper architectures have been shown to achieve state-of-the-art results on many applications [138–140].

However, most of existing multi-layered models for sparse dictionary learning of training data require one to specify a prior of the number of dictionary elements employed within each layer. In many applications, the prior of the number of the dictionary elements is difficult to select, or it is more desirable to infer the number of dictionary elements based on the data itself [140, 141]. The idea of learning an appropriate number and composition of features has motivated the Bayesian nonparametric models, such as Indian buffet process (IBP), as well as Beta-Bernoulli process. Adams *et al.* introduced a cascading Indian buffet process (CIBP), which provided a nonparametric prior on the structure of a layered, directed belief network that was unbounded in both depth and width [141]. However, the problem considered in [141] did not consider multi-layer feature learning [140].

Mittelman *el at.* developed an extension for the restricted Boltzmann machine (RBM) by incorporating a Beta-Bernoulli process factor potential for hidden units. Unlike the standard RBM, this model used the class labels to promote category-dependent sharing of learned features, which tended to improve the generalization performance [142].

Chen *et al.* demonstrated an idea of building an unsupervised deep model that may be cast in terms of a hierarchy of factor-analysis models. The number of canonical dictionary elements or factor loadings at each layer was inferred from the data by an IBP/Beta-Bernoulli construction [140];

As presented above, more flexible and powerful deep multi-layered models combined with non-parametric Bayesian models for deep learning have received attention. We name it *Random Deep Learning* or *Random Multi-layered Models*. As a great success of deep learning has been shown in computer vision, machine learning, artificial intelligence and other areas, we believe random deep learning will present a further progress in this field.

Appendix A

Several Distributions Used in the Thesis

A.1 Gamma Distribution

The Gamma distribution $\Gamma(\alpha, \beta)$ has the probability density function:

$$f(x; \alpha, \beta) = \frac{\alpha^{\beta}}{\Gamma(\beta)} x^{\beta-1} e^{-\alpha x},\ x \in (0, +\infty), \tag{A.1}$$

where $\alpha, \beta > 0$, and $\Gamma(\cdot)$ is the Gamma function.

A.2 Dirichlet Distribution

The Dirichlet distribution $\text{Dir}(x_1, x_2, ..., x_K; \alpha_1, \alpha_2, ..., \alpha_K)$ has the probability density function:

$$f(x_1, x_2, ..., x_K; \alpha_1, \alpha_2, ..., \alpha_K) = \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} x_k^{\alpha_k - 1}, \tag{A.2}$$

where $\alpha_1, \alpha_2, ..., \alpha_K > 0$, $x_1, x_2, ..., x_K > 0$ lies in a K-dimensional simplex, and $\sum_{k=1}^{K} x_k = 1$.

A.3 Beta Distribution

The Beta distribution Beta(α, β) has the probability density function:

$$f(x; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}, \tag{A.3}$$

where $\alpha, \beta > 0$, and $x > 0$.

A.4 Bernoulli Distribution

The Bernoulli distribution for a random variable X has the probability density function:

$$f(x) = \begin{cases} p & \text{if } x = 1 \\ 1-p & \text{if } x = 0 \\ 0 & \text{if } x \notin \{0, 1\}, \end{cases} \tag{A.4}$$

where $p \in (0, 1)$, and $x \in \{0, 1\}$.

A.5 Poisson Distribution

The Poisson distribution for a random variable X has the probability density function:

$$f(X, \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}, \tag{A.5}$$

where $X = 0, 1, 2, ...$, e is Euler's number, and $k!$ is the factorial of k.

A.6 Normal-inverse-Wishart Distribution

The Normal-inverse-Wishart distribution NIW$(\mu, \mathbf{\Sigma}; \nu_0, \lambda, \mathbf{\Psi}, \nu)$ has the probability density function:

$$f(\mu, \mathbf{\Sigma} | \nu_0, \lambda, \mathbf{\Psi}, \nu) = \mathcal{N}(\mu | \mu_\mathbf{0}, \frac{1}{\lambda}\mathbf{\Sigma}) \mathcal{W}^{-1}(\mathbf{\Sigma} | \mathbf{\Psi}, \nu), \tag{A.6}$$

where $\mu_0 \in \mathbb{R}^D$, $\lambda > 0$, $\mathbf{\Psi} \in \mathbb{R}^{D \times D}$ is a covariance matrix, and $\nu > D - 1$.

A.7 Matrix Normal Distribution

The Matrix normal distribution for a random matrix $\mathbf{X}(n \times p)$ has the probability density function:

$$f(\mathbf{X}|\mathbf{M}, \mathbf{U}, \mathbf{V}) = \frac{\exp(-\frac{1}{2}tr[\mathbf{V}^{-1}(\mathbf{X} - \mathbf{M})^T\mathbf{U}^{-1}(\mathbf{X} - \mathbf{M})])}{(2\pi)^{np/2}|\mathbf{V}|^{n/2}|\mathbf{U}|^{p/2}}, \tag{A.7}$$

where tr denotes trace, $\mathbf{M}$ is $n \times p$, $\mathbf{U}$ is $n \times n$ and $\mathbf{V}$ is $p \times p$.

Appendix B

Bayesian Nonparametric Non-negative Matrix Factorization Model and Inference

B.1 Hierarchical Model

The hierarchical form of the model can be expressed as follows:

$$
\begin{aligned}
\mathbf{x_i} =& \mathbf{A}\mathbf{b_i} + e_i \\
\mathbf{b_i} =& \mathbf{z_i} \odot \mathbf{w_i} \\
\mathbf{z_i} \sim& \prod_{k=1}^{K} \text{Bernoulli}(\pi_k) \\
\pi_\mathbf{k} \sim& \text{Beta}(a_0/K, b_0(K-1)/K) \\
A \sim& \prod_{i,k} \varepsilon(A_{i,k}; \alpha_0) \\
\mathbf{w_i} \sim& \prod_{k} \varepsilon(w_{i,k}; \beta_0) \\
e_i \sim& \mathcal{N}(0, \sigma^2) \\
\sigma^2 \sim& \mathcal{G}^{-1}(\kappa, \theta),
\end{aligned}
\tag{B.1}
$$

where $\varepsilon(x;\lambda) = \lambda\text{exp}(-\lambda x)u(x)$ is the exponential density, and $\mathcal{G}^{-1}$ is non-informative inverse gamma prior for the noise variance.

The joint distribution of the model can be expressed as:

$$\begin{aligned} p(\mathbf{X},\mathbf{A},\mathbf{w},\mathbf{Z},\pi,\sigma^2) = &\prod_{i,j}\mathcal{N}(\mathbf{X}_{i,j};(\mathbf{A}(\mathbf{Z}\odot\mathbf{W}))_{i,j},\sigma^2) \\ &\prod_{i,j}\varepsilon(A_{i,j};\alpha_0)\prod_{i,j}\varepsilon(w_{i,j};\beta_0) \\ &\prod_{j}\text{Beta}(\pi_{\mathbf{j}};a_0,b_0) \\ &\prod_{i=1}^{N}\prod_{k=1}^{K}\text{Bernoulli}(z_{ik};\pi_{\mathbf{k}}) \\ &\mathcal{G}^{-1}(\sigma^2;k,\theta) \end{aligned} \tag{B.2}$$

B.2 Gibbs Sampling Inference

To sample $p(\mathbf{A},\mathbf{W},\mathbf{Z},\pi,\sigma^2|\mathbf{X})$, in each iteration of the MCMC, the main steps are shown as follows.

A. Sample A_{ik}:

$$\begin{aligned} p(A_{ik}|\mathbf{X},\mathbf{A}_{\setminus(i,k)},\mathbf{B},\sigma^2) &\propto p(\mathbf{X}|\mathbf{A},\mathbf{B},\sigma^2)p(A_{ik}) \\ &\propto \prod_{i,j}\mathcal{N}(X_{ij};(\mathbf{AB})_{ij},\sigma^2)) \\ &\quad \varepsilon(A_{ik};\alpha_0) \\ &\propto \mathcal{N}(x;\mu_{A_{ik}},\sigma_{A_{ik}^2})\varepsilon(x;\alpha_0). \end{aligned} \tag{B.3}$$

Based on the conjugation between Gaussian distributions, it is easy to get

$$\mu_{A_{ik}} = \frac{\sum_j(\mathbf{x}_{ij} - \sum_{k'\neq k}A_{ik'}B_{k'j})B_{kj}}{\sum_j B_{kj}^2}, \tag{B.4}$$

$$\sigma^2_{A_{ik}} = \frac{\sigma^2}{\sum_j B^2_{kj}}. \tag{B.5}$$

B. Sample $\mathbf{z}_{k:} = [Z_{1k}, Z_{2k}, ..., Z_{Nk}]$:

$$p(Z_{ik}|-) \propto \mathcal{N}(\mathbf{y}_i; \mathbf{A}(\mathbf{w}_i \odot \mathbf{z}_i), \nu^{-1}\mathbf{I})\text{Bernoulli}(Z_{ik}; \pi_k). \tag{B.6}$$

Z_{ik} is equal to either 1 or 0. When $Z_{ik} = 1$, the posterior probability is proportional to

$$p_1 = \pi_k \text{exp}(-\frac{1}{2}\nu(W^2_{ik}\mathbf{a}^T_k\mathbf{a}_k - 2W_{ik}\mathbf{a}^T_k)\widetilde{\mathbf{x}}_i{}^{-k}), \tag{B.7}$$

where $\widetilde{\mathbf{x}}_i{}^{-k} = \mathbf{y}_i - \mathbf{A}(\mathbf{z}_i \odot \mathbf{w}_i) + \mathbf{a}_k(Z_{ik} \odot W_{ik})$.

The posterior probability that $Z_{ik} = 0$ is proportional to

$$p_0 = 1 - \pi_k. \tag{B.8}$$

So Z_{ik} can be drawn from a bernoulli distribution as

$$Z_{ik} \sim \text{Bernoulli}(\frac{p_1}{p_0 + p_1}). \tag{B.9}$$

C. Sample π_k:

$$\pi_k|- \propto \text{Beta}(\pi_k; a_0, b_0) \prod_{i=1}^{N} \text{Bernoulli}(Z_{ik}; \pi_k). \tag{B.10}$$

Because of the conjugate between Beta and Bernoulli distributions, it can be shown that π_k can be drawn from a Beta distribution as

$$\pi_k \sim \text{Beta}(\frac{a_0}{K} + \sum_{i=1}^{N} Z_{ik}, \frac{b_0(K-1)}{K} + N - \sum_{i=1}^{N}(Z_{ik})). \tag{B.11}$$

D. Sample $\mathbf{W}_{kj}$:

Due to symmetry and a similar exponential distribution giving to $\mathbf{W}$ as $\mathbf{A}$, when $\mathbf{B} = \mathbf{Z} \odot \mathbf{W}$, we have

$$p(\mathbf{B}_{kj}|\mathbf{X}, \mathbf{A}, \mathbf{Z}, \mathbf{W}_{\setminus(k,j)}, \sigma^2) \propto \mathcal{N}(x; \mu_{B_{kj}}, \sigma_{B_{kj}^2})\varepsilon(x; \beta_0). \tag{B.12}$$

Notice Z_{kj} is equal to either 1 or 0, therefore, when $Z_{kj} = 0$,

$$p(W_{kj}|\mathbf{X}, \mathbf{A}, \mathbf{B}_{\setminus(k,j)}, \sigma^2) \propto \quad \varepsilon(B_{kj}; \beta_0). \tag{B.13}$$

And when $Z_{kj} = 1$,

$$p(W_{kj})|\mathbf{X}, \mathbf{A}, \mathbf{B}_{\setminus(k,j)}, \sigma^2) \propto \mathcal{N}(x; \mu_{B_{kj}}, \sigma_{B_{kj}^2})\varepsilon(x; \beta_0). \tag{B.14}$$

Finding the elements related to B_{kj}, we have,

$$\mu_{B_{kj}} = \frac{\sum_i (X_{ij} - \sum_{k' \neq k} A_{ik'} B_{k'j}) A_{ik}}{\sum_i A_{ik}^2}, \tag{B.15}$$

$$\sigma_{B_{kj}}^2 = \frac{\sigma^2}{\sum_i B_{kj}^2}. \tag{B.16}$$

E. Sample σ^2:

The conditional density of σ^2 is

$$\begin{aligned} p(\sigma^2|\mathbf{X}, \mathbf{A}, \mathbf{B}) \propto & \prod_{i,j} \mathcal{N}(X_{ij}; (AB)_{ij}, \sigma^2)\mathcal{G}^{-1}(\sigma^2; \kappa, \theta) \\ = & \frac{\theta^k}{\Gamma(k)}(2\pi)^{-i\times j/2}(\sigma^2)^{\frac{-i\times j}{2}-k-1} \\ & \exp(-\frac{1/2\sum_{i,j}(X_{ij} - (AB)_{ij})^2 + \theta}{\sigma^2}) \\ \propto & \mathcal{G}^{-1}(\sigma^2; \kappa_{\sigma^2}, \theta_{\sigma^2}), \end{aligned} \tag{B.17}$$

where $\kappa_{\sigma^2} = \frac{i\times j}{2} + k$, and $\theta_{\sigma^2} = \frac{1}{2}\sum_{i,j}(X - AB)_{i,j}^2 + \theta$.

Bibliography

[1] Peter Orbanz and Yee Whye Teh. Bayesian nonparametric models. In *Encyclopedia of Machine Learning*, pages 81–89. Springer, 2011.

[2] Mingyuan Zhou. *Nonparametric bayesian dictionary learning and count and mixture modeling*. PhD thesis, Duke University, 2013.

[3] Kai Mao. *Nonparametric Bayesian Models for Supervised Dimension Reduction and Regression*. PhD thesis, Duke University, 2009.

[4] Scott Niekum. A brief introduction to bayesian nonparametric methods for clustering and time series analysis. *Technical report CMU-RI-TR-15-02, Robotics Institute, Carnegie Mellon University*.

[5] John Paisley. *Machine learning with Dirichlet and beta process priors: Theory and Applications*. PhD thesis, Duke University, 2010.

[6] Emily B Fox. *Bayesian nonparametric learning of complex dynamical phenomena*. PhD thesis, Massachusetts Institute of Technology, 2009.

[7] Xuhui Fan. *Nonparametric Bayesin Models for Learning Network Coupling Relationships*. PhD thesis, University of Technology, Sydney, 2015.

[8] Peter Orbanz and Joachim M Buhmann. Smooth image segmentation by nonparametric bayesian inference. In *Computer Vision–ECCV 2006*, pages 444–457. Springer, 2006.

[9] Soumya Ghosh, Andrei B Ungureanu, Erik B Sudderth, and David M Blei. Spatial distance dependent chinese restaurant processes for image segmentation. In *Advances in Neural Information Processing Systems*, pages 1476–1484, 2011.

[10] Peter Orbanz and Joachim M Buhmann. Nonparametric bayesian image segmentation. *International Journal of Computer Vision*, 77(1-3):25–45, 2008.

[11] David M Blei and Peter I Frazier. Distance dependent chinese restaurant processes. *The Journal of Machine Learning Research*, 12:2461–2488, 2011.

[12] EB Sudderth and S Ghosh. Nonparametric learning for layered segmentation of natural images. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2272–2279. IEEE, 2012.

[13] Alex Shyr, Trevor Darrell, Michael Jordan, and Raquel Urtasun. Supervised hierarchical pitman-yor process for natural scene segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2281–2288. IEEE, 2011.

[14] John Paisley and Lawrence Carin. Nonparametric factor analysis with beta process priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 777–784. ACM, 2009.

[15] John Paisley, Mingyuan Zhou, Guillermo Sapiro, and Lawrence Carin. Nonparametric image interpolation and dictionary learning using spatially-dependent dirichlet and beta process priors. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 1869–1872. IEEE, 2010.

[16] Mingyuan Zhou, Haojun Chen, John Paisley, Lu Ren, Lingbo Li, Zhengming Xing, David Dunson, Guillermo Sapiro, and Lawrence Carin. Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images. *Image Processing, IEEE Transactions on*, 21(1):130–144, 2012.

[17] Raied Aljadaany, Khoa Luu, Shreyas Venugopalan, and Marios Savvides. Iris super-resolution via nonparametric over-complete dictionary learning. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 3856–3860. IEEE, 2015.

[18] Gungor Polatkan, MengChu Zhou, Lawrence Carin, David Blei, and Ingrid Daubechies. A bayesian nonparametric approach to image super-resolution. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(2):346–358, 2015.

[19] Emily B Fox, Michael I Jordan, Erik B Sudderth, and Alan S Willsky. Sharing features among dynamical systems with beta processes. In *Advances in Neural Information Processing Systems*, pages 549–557, 2009.

[20] Ava Bargi, M Piccardi, and Zoubin Ghahramani. A non-parametric conditional factor regression model for multi-dimensional input and response. In *17th International Conference on Artificial Intelligence and Statistics*, 2014.

[21] David Knowles and Zoubin Ghahramani. Infinite sparse factor analysis and infinite independent components analysis. In *Independent Component Analysis and Signal Separation*, pages 381–388. Springer, 2007.

[22] Erik B Sudderth. *Graphical models for visual object recognition and tracking*. PhD thesis, Massachusetts Institute of Technology, 2006.

[23] John Paisley and Lawrence Carin. A nonparametric bayesian model for kernel matrix completion. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 2090–2093. IEEE, 2010.

[24] Mingyuan Zhou, Chunping Wang, Minhua Chen, et al. Nonparametric bayesian matrix completion. In *2010 IEEE Sensor Array and Multichannel Signal Processing Workshop*, pages 213–216, 2010.

[25] Christos E Kountzakis. Random measures do produce generalized likelihoods. 2014.

[26] Alan F Karr et al. Lévy random measures. *The Annals of Probability*, 6(1):57–71, 1978.

[27] John Kingman. Completely random measures. *Pacific Journal of Mathematics*, 21(1): 59–78, 1967.

[28] Rajesh Ranganath and David Blei. Correlated random measures. *arXiv preprint arXiv:1507.00720*, 2015.

[29] Wikipedia. *Random measure*, 2014 (accessed January 12, 2015). URL https://en.wikipedia.org/wiki/Random_measure.

[30] Alan Karr. *Point processes and their statistical inference*, volume 7. CRC press, 1991.

[31] Ç Erhan et al. *Probability and stochastics*, volume 261. Springer Science & Business Media, 2011.

[32] Peter Tankov. Financial modeling with lévy processes. *Lecture notes*, 2010.

[33] Zenghu Li. *Measure-valued branching Markov processes*. Springer Science & Business Media, 2010.

[34] Michael I Jordan. Hierarchical models, nested models and completely random measures. *Frontiers of Statistical Decision Making and Bayesian Analysis: in Honor of James O. Berger. New York: Springer*, 2010.

[35] Peter Orbanz and Sinead Williamson. Unit–rate poisson representations of completely random measures. Technical report, 2011.

[36] Kazuhisa Matsuda. Introduction to the mathematics of lévy processes. *A part of Ph. D. thesis*, 2005.

[37] Delia Coculescu and Ashkan Nikeghbali. Filtrations. *Encyclopedia of Quantitative Finance*, pages 2461–2488, 2010.

[38] Antonis Papapantoleon. An introduction to lévy processes with applications in finance. *arXiv preprint arXiv:0804.0482*, 2008.

[39] Andreas E. Kyprianou. *Lévy processes and continuous state branching processes: part I*, 2008 (accessed February 12, 2015). URL http://www.maths.bath.ac.uk/~ak257/LevyCSB.html.

[40] K Sato. *Lévy processes and infinitely divisible distributions*. Cambridge university press, 1999.

[41] Andreas Eberle. Stochastic analysis, 2013.

[42] Andreas Kyprianou. *Introductory lectures on fluctuations of Lévy processes with applications*. Springer Science & Business Media, 2006.

[43] Andreas Kyprianou. *Fluctuations of Lévy processes with applications: Introductory Lectures*. Springer Science & Business Media, 2014.

[44] Yingjian Wang and Lawrence Carin. Levy measure decompositions for the beta and gamma processes. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 73–80, 2012.

[45] Yun Zhang. *Wiener and Gamma Process Overview for Degradation Modelling and Prognostic*. PhD thesis, Norwegian University of Science and Technology, 2009.

[46] Mark Senn. *gamma process*, 2014 (accessed January 12, 2015). URL https://en.wikipedia.org/wiki/Gamma_process.

[47] H Jeffreys and BS Jeffreys. Frullani's integrals. *Methods of Mathematical Physics, 3rd ed., Cambridge University Press, Cambridge, UK*, pages 406–407, 1988.

[48] Yee Whye Teh. Dirichlet process. In *Encyclopedia of machine learning*, pages 280–287. Springer, 2010.

[49] Xinhua Zhang. A very gentle note on the construction of dirichlet process. *The Australian National University, Canberra*, 2008.

[50] Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

[51] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. *Journal of the american statistical association*, 2012.

[52] Samuel J Gershman and David M Blei. A tutorial on bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1–12, 2012.

[53] John W Paisley, Aimee K Zaas, Christopher W Woods, Geoffrey S Ginsburg, and Lawrence Carin. A stick-breaking construction of the beta process. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 847–854, 2010.

[54] Lawrence Carin, David M Blei, and John W Paisley. Variational inference for stick-breaking beta process priors. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 889–896, 2011.

[55] John W Paisley, David M Blei, and Michael I Jordan. Stick-breaking beta processes and the poisson process. In *International Conference on Artificial Intelligence and Statistics*, pages 850–858, 2012.

[56] Tamara Broderick, Michael I Jordan, Jim Pitman, et al. Beta processes, stick-breaking and power laws. *Bayesian analysis*, 7(2):439–476, 2012.

[57] Romain Thibaux and Michael I Jordan. Hierarchical beta processes and the indian buffet process. In *International conference on artificial intelligence and statistics*, pages 564–571, 2007.

[58] Wikipedia. *Bernoulli process*, 2011 (accessed January 12, 2014). URL https://www.wikiwand.com/en/Bernoulli_process.

[59] Thomas L Griffiths and Zoubin Ghahramani. The indian buffet process: An introduction and review. *The Journal of Machine Learning Research*, 12:1185–1224, 2011.

[60] Zoubin Ghahramani and Thomas L Griffiths. Infinite latent feature models and the indian buffet process. In *Advances in neural information processing systems*, pages 475–482, 2005.

[61] Yingjian Wang and Lawrence Carin. Levy measure decompositions for the beta and gamma processes. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 73–80. IEEE, 2012.

[62] Dustin Stansbury. *A Brief Introduction to Markov Chains*, 2012 (accessed February 6, 2014). URL https://theclevermachine.wordpress.com/2012/09/24/a-brief-introduction-to-markov-chains/.

[63] Wikipedia. *Markov chain*, 2016 (accessed February 6, 2016). URL https://en.wikipedia.org/wiki/Markov_chain.

[64] Dustin Stansbury. *MCMC: The Metropolis-Hastings Sampler*, 2012 (accessed February 6, 2014). URL https://theclevermachine.wordpress.com/2012/10/20/mcmc-the-metropolis-hastings-sampler/.

[65] Dustin Stansbury. *MCMC: The Gibbs Sampler*, 2012 (accessed February 6, 2014). URL https://theclevermachine.wordpress.com/2012/10/20/mcmc-the-gibbs-sampler/.

[66] Tim Hesterberg. Monte carlo strategies in scientific computing. *Technometrics*, 44(4): 403–404, 2002.

[67] Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[68] Robert E Kass, Bradley P Carlin, Andrew Gelman, and Radford M Neal. Markov chain monte carlo in practice: a roundtable discussion. *The American Statistician*, 52(2):93–100, 1998.

[69] Nannan Wang, Dacheng Tao, Xinbo Gao, Xuelong Li, and Jie Li. A comprehensive survey to face hallucination. *International Journal of Computer Vision*, 106(1):9–30, 2014.

[70] Xiaoguang Li, Qing Xia, Li Zhuo, and Kin Man Lam. A face hallucination algorithm via kpls-eigentransformation model. In *Signal Processing, Communication and Computing (ICSPCC), 2012 IEEE International Conference on*, pages 462–467. IEEE, 2012.

[71] Ce Liu, Heung-Yeung Shum, and William T Freeman. Face hallucination: Theory and practice. *International Journal of Computer Vision*, 75(1):115–134, 2007.

[72] Kaibing Zhang, Xinbo Gao, Xuelong Li, and Dacheng Tao. Partially supervised neighbor embedding for example-based image super-resolution. *Selected Topics in Signal Processing, IEEE Journal of*, 5(2):230–239, 2011.

[73] Chih-Chung Hsu, Chia-Wen Lin, Chiou-Ting Hsu, H-YM Liao, and Jen-Yu Yu. Face hallucination using bayesian global estimation and local basis selection. In *Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on*, pages 449–453. IEEE, 2010.

[74] Raanan Fattal. Image upsampling via imposed edge statistics. In *ACM Transactions on Graphics (TOG)*, volume 26, page 95. ACM, 2007.

[75] Jian Sun, Nan-Ning Zheng, Hai Tao, and Heung-Yeung Shum. Image hallucination with primal sketch priors. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–729. IEEE, 2003.

[76] Jian Sun, Jian Sun, Zongben Xu, and Heung-Yeung Shum. Image super-resolution using gradient profile prior. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[77] Jian Sun, Zongben Xu, and Heung-Yeung Shum. Gradient profile prior and its applications in image super-resolution and enhancement. *Image Processing, IEEE Transactions on*, 20(6):1529–1542, 2011.

[78] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. Super-resolution through neighbor embedding. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2004.

[79] Jianchao Yang, Hao Tang, Yi Ma, and Thomas Huang. Face hallucination via sparse coding. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1264–1267. IEEE, 2008.

[80] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma. Image super-resolution as sparse representation of raw image patches. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[81] Lyndsey C Pickup, David P Capel, Stephen J Roberts, and Andrew Zisserman. Bayesian image super-resolution, continued. In *Advances in Neural Information Processing Systems*, pages 1089–1096, 2006.

[82] Chih-Yuan Yang and Ming-Hsuan Yang. Fast direct super-resolution by simple functions. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 561–568. IEEE, 2013.

[83] Chih-Yuan Yang, Sifei Liu, and Ming-Hsuan Yang. Structured face hallucination. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1099–1106. IEEE, 2013.

[84] Cheolkon Jung, Licheng Jiao, Bing Liu, and Maoguo Gong. Position-patch based face hallucination using convex optimization. *Signal Processing Letters, IEEE*, 18(6):367–370, 2011.

[85] Katherine Donaldson and Gregory K Myers. Bayesian super-resolution of text in videowith a text-specific bimodal prior. *International Journal of Document Analysis and Recognition (IJDAR)*, 7(2-3):159–167, 2005.

[86] Jing Tian and Kai-Kuang Ma. A mcmc approach for bayesian super-resolution image reconstruction. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 1, pages I–45. IEEE, 2005.

[87] Soumya Ghosh, Matthew Loper, Erik B Sudderth, and Michael J Black. From deformations to parts: Motion-based segmentation of 3d objects. In *Advances in Neural Information Processing Systems*, pages 1997–2005, 2012.

[88] Michael Elad. *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer, 2010.

[89] Emily Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. Nonparametric bayesian learning of switching linear dynamical systems. In *Advances in Neural Information Processing Systems*, pages 457–464, 2009.

[90] Ferdinando S Samaria and Andy C Harter. Parameterisation of a stochastic model for human face identification. In *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*, pages 138–142. IEEE, 1994.

[91] Peter N Belhumeur, João P Hespanha, and David J Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):711–720, 1997.

[92] Patrik O Hoyer. Non-negative sparse coding. In *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, pages 557–565. IEEE, 2002.

[93] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[94] Yu-Xiong Wang and Yu-Jin Zhang. Nonnegative matrix factorization: A comprehensive review. *Knowledge and Data Engineering, IEEE Transactions on*, 25(6):1336–1353, 2013.

[95] Patrik Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004.

[96] Jingu Kim and Haesun Park. Toward faster nonnegative matrix factorization: A new algorithm and comparisons. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 353–362. IEEE, 2008.

[97] Mikkel N Schmidt and Hans Laurberg. Nonnegative matrix factorization with gaussian process priors. *Computational intelligence and neuroscience*, 2008:3, 2008.

[98] Nan Ding, Rongjing Xiang, Ian Molloy, Ninghui Li, et al. Nonparametric bayesian matrix factorization by power-ep. In *International Conference on Artificial Intelligence and Statistics*, pages 169–176, 2010.

[99] Michael W Berry, Murray Browne, Amy N Langville, V Paul Pauca, and Robert J Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational statistics & data analysis*, 52(1):155–173, 2007.

[100] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

[101] V Paul Pauca, Jon Piper, and Robert J Plemmons. Nonnegative matrix factorization for spectral data analysis. *Linear algebra and its applications*, 416(1):29–47, 2006.

[102] Zhe Chen and Andrzej Cichocki. Nonnegative matrix factorization with temporal smoothness and/or spatial decorrelation constraints. *Laboratory for Advanced Brain Signal Processing, RIKEN, Tech. Rep*, 68, 2005.

[103] Nancy Bertin, Roland Badeau, and Emmanuel Vincent. Enforcing harmonicity and smoothness in bayesian non-negative matrix factorization applied to polyphonic music transcription. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(3): 538–549, 2010.

[104] Mikkel N Schmidt, Ole Winther, and Lars Kai Hansen. Bayesian non-negative matrix factorization. In *Independent Component Analysis and Signal Separation*, pages 540–547. Springer, 2009.

[105] Said Moussaoui, David Brie, Ali Mohammad-Djafari, and Cédric Carteret. Separation of non-negative mixture of non-negative sources using a bayesian approach and mcmc sampling. *Signal Processing, IEEE Transactions on*, 54(11):4133–4145, 2006.

[106] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1548–1560, 2011.

[107] Liansheng Zhuang, Haoyuan Gao, Zhouchen Lin, Yi Ma, Xin Zhang, and Nenghai Yu. Non-negative low rank and sparse graph for semi-supervised learning. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2328–2335. IEEE, 2012.

[108] Bin Cheng, Jianchao Yang, Shuicheng Yan, Yun Fu, and Thomas S Huang. Learning with-graph for image analysis. *Image Processing, IEEE Transactions on*, 19(4):858–866, 2010.

[109] Mingyuan Zhou, Haojun Chen, Lu Ren, Guillermo Sapiro, Lawrence Carin, and John W Paisley. Non-parametric bayesian dictionary learning for sparse image representations. In *Advances in Neural Information Processing Systems*, pages 2295–2303, 2009.

[110] Tao Li and Chris HQ Ding. Nonnegative matrix factorizations for clustering: A survey. pages 149–176, 2013.

[111] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[112] Hong Cheng, Zicheng Liu, Lu Yang, and Xuewen Chen. Sparse representation and learning in visual recognition: theory and applications. *Signal Processing*, 93(6):1408–1425, 2013.

[113] Yongjiao Wang, Chuan Wang, and Lei Liang. Sparse representation theory and its application for face recognition. *International Journal on Smart Sensing & Intelligent Systems*, 8(1), 2015.

[114] John Wright, Yi Ma, Julien Mairal, Guillermo Sapiro, Thomas S Huang, and Shuicheng Yan. Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, 98(6):1031–1044, 2010.

[115] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227, 2009.

[116] Lei Zhang, Meng Yang, Zhizhao Feng, and David Zhang. On the dimensionality reduction for sparse representation based face recognition. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1237–1240. IEEE, 2010.

[117] Haichao Zhang, Yanning Zhang, and Thomas S Huang. Pose-robust face recognition via sparse representation. *Pattern Recognition*, 46(5):1511–1521, 2013.

[118] Yong Xu, Qi Zhu, Yan Chen, Jeng-Shyang Pan, et al. An improvement to the nearest neighbor classifier and face recognition experiments. *Int J Innov Comput Inf Control*, 8 (12):1349–4198, 2012.

[119] Ke Huang and Selin Aviyente. Sparse representation for signal classification. In *Advances in neural information processing systems*, pages 609–616, 2006.

[120] Zheng Zhang, Yong Xu, Jian Yang, Xuelong Li, and David Zhang. A survey of sparse representation: algorithms and applications. *Access, IEEE*, 3:490–530, 2015.

[121] Meng Yang, Lei Zhang, Jian Yang, and David Zhang. Robust sparse coding for face recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 625–632. IEEE, 2011.

[122] Allen Y Yang, John Wright, Yi Ma, and S Shankar Sastry. Feature selection in face recognition: A sparse representation perspective. *submitted to Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2007.

[123] Long Ma, Chunheng Wang, Baihua Xiao, and Wen Zhou. Sparse representation for face recognition based on discriminative low-rank dictionary learning. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2586–2593. IEEE, 2012.

[124] Lei Zhang, Meng Yang, Xiangchu Feng, Yi Ma, and David Zhang. Collaborative representation based classification for face recognition. *arXiv preprint arXiv:1204.2358*, 2012.

[125] Qinfeng Shi, Anders Eriksson, Anton Van Den Hengel, and Chunhua Shen. Is face recognition really a compressive sensing problem? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 553–560. IEEE, 2011.

[126] Lei Zhang, Meng Yang, and Xiangchu Feng. Sparse representation or collaborative representation: Which helps face recognition? In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 471–478. IEEE, 2011.

[127] Lingfeng Wang, Huaiyu Wu, and Chunhong Pan. Manifold regularized local sparse representation for face recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 25(4):651–659, 2015.

[128] Liansheng Zhuang, Shenghua Gao, Jinhui Tang, Jingjing Wang, Zhouchen Lin, Yi Ma, and Nenghai Yu. Constructing a nonnegative low-rank and sparse graph with data-adaptive features. *Image Processing, IEEE Transactions on*, 24(11):3717–3728, 2015.

[129] Aleix M Martinez. The ar face database. *CVC Technical Report*, 24, 1998.

[130] Claus Skaanning Jensen. *Blocking Gibbs sampling for inference in large and complex Bayesian networks with applications in genetics*. PhD thesis, Aalborg University, 1997.

[131] Jason Chang and John W Fisher III. Parallel sampling of dp mixture models using sub-cluster splits. In *Advances in Neural Information Processing Systems*, pages 620–628, 2013.

[132] Xiangyu Wang, Fangjian Guo, Katherine A Heller, and David B Dunson. Parallelizing mcmc with random partition trees. In *Advances in Neural Information Processing Systems*, pages 451–459, 2015.

[133] Deepak Venugopal and Vibhav Gogate. Dynamic blocking and collapsing for gibbs sampling. *arXiv preprint arXiv:1309.6870*, 2013.

[134] Olav Kallenberg. *Probabilistic symmetries and invariance principles*. Springer Science & Business Media, 2006.

[135] James Lloyd, Peter Orbanz, Zoubin Ghahramani, and Daniel M Roy. Random function priors for exchangeable arrays with applications to graphs and relational data. In *Advances in Neural Information Processing Systems*, pages 1007–1015, 2012.

[136] Diana Cai, Nathanael Ackerman, and Cameron Freer. Priors on exchangeable directed graphs. *arXiv preprint arXiv:1510.08440*, 2015.

[137] Francois Caron and Emily B Fox. Sparse graphs using exchangeable random measures. *arXiv preprint arXiv:1401.1137*, 2014.

[138] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[139] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[140] Bo Chen, Gungor Polatkan, Guillermo Sapiro, David Blei, David Dunson, and Lawrence Carin. Deep learning with hierarchical convolutional factor analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1887–1901, 2013.

[141] Ryan Prescott Adams, Hanna M Wallach, and Zoubin Ghahramani. Learning the structure of deep sparse graphical models. *arXiv preprint arXiv:1001.0160*, 2009.

[142] Roni Mittelman, Honglak Lee, Benjamin Kuipers, and Silvio Savarese. Weakly supervised learning of mid-level features with beta-bernoulli process restricted boltzmann machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 476–483, 2013.