Machine Learning with Dirichlet and Beta Process Priors: Theory and Applications

by

John Paisley

Department of Electrical & Computer Engineering
Duke University

Date: _____

Approved:

_____
Lawrence Carin, Advisor

_____
David Brady

_____
Rebecca Willett Lu

_____
David Dunson

_____
Mauro Maggioni

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Electrical & Computer Engineering in the Graduate School of Duke University
2010

Abstract

(Machine Learning)

Machine Learning with Dirichlet and Beta Process Priors: Theory and Applications

by

John Paisley

Department of Electrical & Computer Engineering
Duke University

Date: _____

Approved:

_____
Lawrence Carin, Advisor

_____
David Brady

_____
Rebecca Willett Lu

_____
David Dunson

_____
Mauro Maggioni

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Electrical & Computer Engineering
in the Graduate School of Duke University
2010

Abstract

Bayesian nonparametric methods are useful for modeling data without having to define the complexity of the entire model *a priori*, but rather allowing for this complexity to be determined by the data. Two problems considered in this dissertation are the number of components in a mixture model, and the number of factors in a latent factor model, for which the Dirichlet process and the beta process are the two respective Bayesian nonparametric priors selected for handling these issues.

The flexibility of Bayesian nonparametric priors arises from the prior's definition over an infinite dimensional parameter space. Therefore, there are theoretically an *infinite* number of latent components and an *infinite* number of latent factors. Nevertheless, draws from each respective prior will produce only a small number of components or factors that appear in a given data set. As mentioned, the number of these components and factors, and their corresponding parameter values, are left for the data to decide.

This dissertation is split between novel practical applications and novel theoretical results for these priors. For the Dirichlet process, we investigate stick-breaking representations for the finite Dirichlet process and their application to novel sampling techniques, as well as a novel mixture modeling framework that incorporates multiple modalities within a data set. For the beta process, we present a new stick-breaking construction for the infinite-dimensional prior, and consider applications to image interpolation problems and dictionary learning for compressive sensing.

Contents

List of Tables

List of Figures

Acknowledgements

I am very grateful to my advisor Lawrence Carin for all of his help and guidance in my pursuit of the PhD. I would also like to thank William Joines, without whom I would not be in this position in the first place. I thank David Dunson and Mauro Maggioni for their constructive criticisms on my work over the past four years. I also appreciate the help of David Blei at Princeton University, whose conversations have lead to additions that have strengthened this dissertation.

I thank my parents, Scott and Jackie Paisley, for their support throughout my life in my academic pursuits, and my sisters Jenny and Sarah for keeping me grounded.

Finally, I would like to thank all of my tóngxué for their friendships and for bringing half of the world to my attention.

Introduction

Bayesian nonparametric methods are useful for modeling data without having to define the complexity of the entire model *a priori*, but rather allowing for this complexity to be determined by the data. Two problems considered in this dissertation are the number of components in a mixture model, and the number of factors in a latent factor model, for which the Dirichlet process and the beta process are the two respective Bayesian nonparametric priors selected for handling these issues.

The flexibility of Bayesian nonparametric priors arises from the prior's definition over an infinite dimensional parameter space. Therefore, there are theoretically an *infinite* number of latent components and an *infinite* number of latent factors. Nevertheless, draws from each respective prior will produce only a small number of components or factors that appear in a given data set. As mentioned, the number of these components and factors, and their corresponding parameter values, are left for the data to decide.

Below, we briefly give three examples of problems that motivate Bayesian nonparametric methods and clearly illustrate their utility. These examples concern three standard modeling problems: (*i*) the Gaussian mixture model [9], (*ii*) the hidden Markov model [61][9] and (*iii*) the matrix factorization problem [55] (and references therin). These last two examples are taken from research published by the author, but not included in this dissertation.

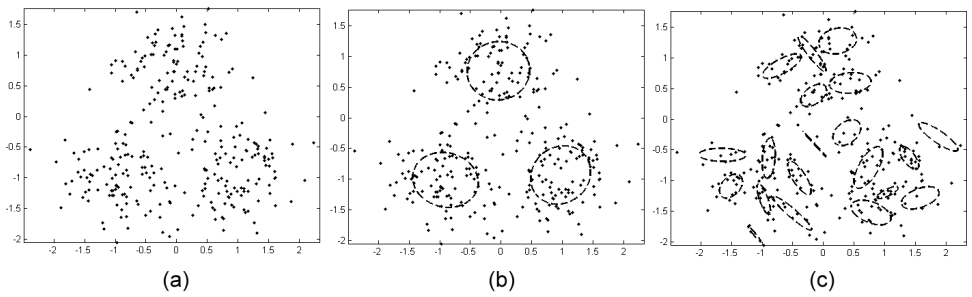


FIGURE 1: (a) The original data set. (b) Clustering results for a Gaussian mixture model using a sparsity-promoting Dirichlet prior on the mixing weights and learned using VB-EM inference. Of the initial 20 components, only 3 are ultimately used and shown. (c) Clustering results for a Gaussian mixture model using the maximum likelihood EM algorithm to learn the model parameters. All 20 components are used by the data, resulting in clear overfitting.

The Gaussian Mixture Model

The Gaussian mixture model (GMM) models vectors $x \in \mathbb{R}^d$ in a data set, $\{x_n\}_{n=1}^N$, as being generated according to the distribution

$$x_n \overset{iid}{\sim} \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \tag{1}$$

The vector $(\pi_1, \ldots, \pi_K)$ consists of probability weights and sum to one. The value π_k gives the probability that observation x_n is generated from a Gaussian with mean vector μ_k and covariance matrix Σ_k.

The classical approach to learning the parameters $\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$ is by maximum likelihood using the EM algorithm [21]. In this algorithm, the value of K is fixed and the algorithm iterates between updating the distribution of a latent indicator variable, and updating the parameters to maximize the likelihood given the soft clustering induced by this latent indicator. In Figure 1, we show how this can lead to overfitting if K is selected poorly. Extending the GMM to the Bayesian realm, Figure 1 shows a clear advantage. The prior on π is the Dirichlet distribution and will be discussed in detail in this dissertation.

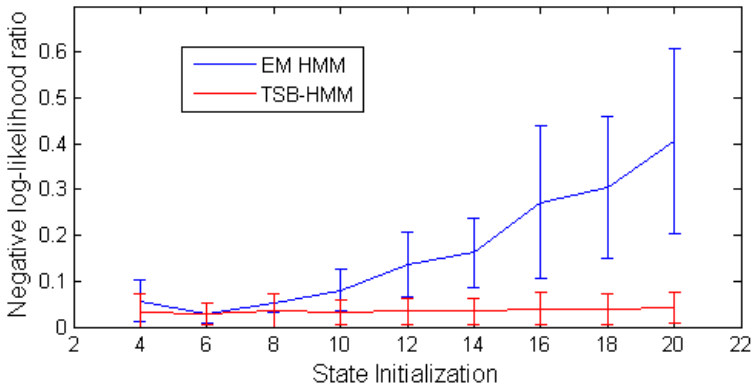


FIGURE 2: The empirical Kullback-Liebler divergence between the true underlying HMM and the learned HMM using the ML-EM algorithm (blue) and the fully Bayesian VB-EM algorithm (red). This figure is taken from [52].

The Hidden Markov Model

Hidden markov models (HMM) are useful for modeling sequential data. They model a sequence, $(X_1, X_2, \dots)$, as being generated from a distribution whose parameters are selected by a hidden sequence of indicators, $(S_1, S_2, \dots)$. This hidden sequence is a Markov chain and the number of states in this chain, K, is typically set *a priori*.

The classical approach to learning the transition probabilities and state-dependent parameter values is by maximum likelihood using the EM algorithm [61]. As with the GMM, this iterates between learning distributions on the latent state indicators, followed by an updating of all parameters to maximize the likelihood. In Figure 2, we show a comparison of the maximum likelihood approach with the Bayesian approach as a function of the initialization of the number of states. We generated sequences from a 4-state discrete HMM and used both methods to learn the underlying HMM given these sequences. The plot contains an empirical approximation of the Kullback-Liebler divergence [33] between the true model and the inferred model. As is evident, the ML approach is sensitive to the number of states, while the Bayesian approach sets the probabilities of all superfluous states to zero.

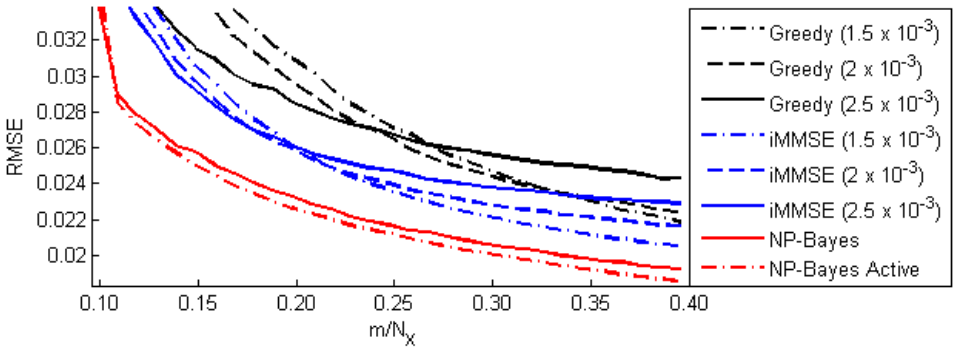


FIGURE 3: The RMSE of the interpolated matrix values to the true values for a matrix completion problem where the symmetric matrix is modeled as $X = \Phi^T\Phi + E$. The x-axis is a function of increasing number of measurements. This figure is taken from [55].

Matrix Factorization Models

As a final example, we consider learning factorizations of incomplete matrices with the goal of completing them. Consider the positive semidefinite matrix, X, and a low rank factorization $X = \Phi^T\Phi + E$, where E accounts for errors due to smaller eigenvalues. When many values of X are missing, this factorization cannot be learned via an eigendecomposition, and so Φ must be learned using an algorithm. Two least squares methods for solving this problem consist of gradually increasing the rank of the factorization, and then minimizing the squared error to the measured values. For each increase, all values in Φ can change (iMMSE), or only the values in the added dimension (greedy) [51]. A Bayesian method is to model the columns of Φ with a sparsity-promoting normal-gamma prior, where the gamma-distributed precision values are shared among the columns of Φ. This learns the proper rank by squeezing out all unnecessary dimensions. Theoretically, the dimensionality of Φ is infinite, but in practice the number of nonzero dimensions will be only that required by the data. In Figure 3, we show results where the matrix X is a 2250×2250 Gaussian kernel that measures similarity between pieces of music.

This dissertation is organized as follows:

- **Chapter 1**: In Chapter 1, we review the Dirichlet process in detail. This includes derivations of the two common representations of this infinite-dimensional prior. We end the chapter by calculating the expectation and variance of the entropy of Dirichlet processes. To our knowledge, this is a new calculation and is an addition to the theoretical properties of measures drawn from Dirichlet process priors.

- **Chapter 2**: In Chapter 2, we continue the discussion of Chapter 1 by looking more in-depth at Sethuraman's stick-breaking construction of a finite-dimensional Dirichlet prior [68]. This includes a novel comparison of the finite-symmetric Dirichlet distribution and the truncated Dirichlet process as mixture modeling priors, and two new applications of this construction for (i) performing conjugate inference for the concentration parameter of a Dirichlet distribution and (ii) conjugate inference for the hierarchical Dirichlet process [70].

- **Chapter 3**: In Chapter 3, we extend the framework of Dirichlet process priors to include data that has multiple modalities. This simple modification allows for mixture modeling to be performed jointly on multiple aspects of a data set. A novel application of this framework is discussed later in Chapter 6.

- **Chapter 4**: In Chapter 4, we shift attention to the beta process for Bayesian nonparametric learning of latent factor models. This can also be cast as a nonparametric matrix factorization model, or a method for nonparametric dictionary learning. We give a general review of the problem, followed by a new variational inference algorithm for model learning. We apply the model to several data sets, and also consider a compressive sensing application, where the

beta process is used to nonparametrically learn an overcomplete dictionary to be used as a nonorthogonal basis for CS inversion.

- **Chapter 5**: In Chapter 5, we present a new stick-breaking construction of the beta process. We believe this is a major theoretical contribution to the theory of beta processes, as the stick-breaking construction of the Dirichlet process was to Dirichlet processes. We give a proof of the construction, as well as a method for performing inference for this prior. We apply the prior to several data sets, including a time-evolving gene data set.

- **Chapter 6**: In Chapter 6, we incorporate the model of Chapter 4 into the framework presented in Chapter 3 by presenting new models for image interpolation. The central model uses two modalities, the second of which acts as a spatial prior. Images considered include natural (rgb) images, and hyperspectral images.

1

The Dirichlet Process for Mixture Models

1.1 Abstract

In this chapter, we review the Dirichlet process beginning with a review of the finite-dimensional Dirichlet distribution and its representations as a Pólya Urn process and an inifinte stick-breaking process. We then briefly extend these ideas to infinite-dimensional spaces. In Section 1.7, we calculate the expectation and variance of the entropy of probability measures drawn from the Dirichlet process prior. To our knowledge, this calculation is a new contribution to the theory of Dirichlet processes.

1.2 The Dirichlet Distribution

Consider the finite, K-dimensional vector, $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$, where $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$. Vectors of this form are said to reside in the $(K-1)$-dimensional simplex of $\mathbb{R}^K$, denoted $\boldsymbol{\pi} \in \Delta_K$. We view this vector as the parameter for the multinomial distribution, where samples $X \sim \text{Mult}(\{1, \ldots, K\}, \boldsymbol{\pi})$ take values $X \in \{1, \ldots, K\}$ with the probability that $X = k$ equal to π_k. When the vector $\boldsymbol{\pi}$ is unknown, it can be inferred in the Bayesian setting by using its conjugate prior, the

Dirichlet distribution.

The Dirichlet distribution of dimension K is a continuous probability distribution on Δ_K and has the density function

$$p(\boldsymbol{\pi}|\beta_1, \ldots, \beta_K) = \frac{\Gamma(\sum_k \beta_k)}{\prod_k \Gamma(\beta_k)} \prod_{k=1}^{K} \pi_k^{\beta_k - 1} \tag{1.1}$$

where the parameters $\beta_k \geq 0$, $\forall k$. It is useful to reparameterize this distribution by defining $\alpha := \sum_k \beta_k$ and the vector $g_0 \in \Delta_K$, with $g_{0k} := \beta_k / \sum_k \beta_k$.

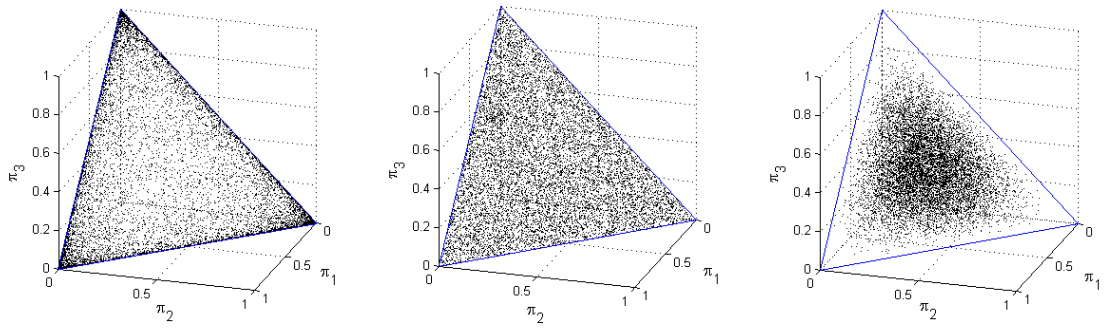
$$p(\boldsymbol{\pi}|\alpha g_{01}, \ldots, \alpha g_{0K}) = \frac{\Gamma(\alpha)}{\prod_i \Gamma(\alpha g_{0i})} \prod_{i=1}^{K} \pi_i^{\alpha g_{0i} - 1} \tag{1.2}$$

A vector with this distribution is denoted $\boldsymbol{\pi} \sim \text{Dir}(\alpha g_0)$. The mean and variance of an element in $\boldsymbol{\pi}$ is

$$\mathbb{E}[\pi_k|\alpha g_0] = g_{0k}, \qquad \mathbb{V}[\pi_k|\alpha g_0] = \frac{g_{0k}(1 - g_{0k})}{\alpha(\alpha + 1)} \tag{1.3}$$

Therefore, g_0 functions as a prior guess of $\boldsymbol{\pi}$ and α as a strength parameter, controlling how tight the distribution is around g_0. When $g_{0k} = 0$, $\pi_i = 0$ with probability one, and when $g_{0i} = 1$ and $g_{0k} = 0$, $\forall k \neq i$, $\boldsymbol{\pi} = \mathbf{e}_i$ with probability one, where $\mathbf{e}_i$ is a vector of zeros, except for a one in the i^{th} position.

Figure 1.1 shows plots, each containing 10,000 samples drawn from a 3-dimensional Dirichlet distribution with g_0 uniform and $\alpha = 1, 3, 10$. This gives insight into the function of α. When $\alpha = K$, or the dimensionality of the Dirichlet distribution, we see that the density is uniform on the simplex; when $\alpha > K$, the density begins to cluster around g_0. Perhaps more interesting, and more relevant to the Dirichlet process, is when $\alpha < K$. We see that as α becomes less than the dimensionality of the distribution, most of the density lies on the corners and faces of the simplex. In general, as the ratio of α to K shrinks, draws of $\boldsymbol{\pi}$ will be *sparse*, meaning that



(a) (b) (c)

FIGURE 1.1: 10,000 samples from a 3-dimensional Dirichlet distribution with g_0 uniform and (a) $\alpha = 1$ (b) $\alpha = 3$ (c) $\alpha = 10$ As can be seen, (a) when $\alpha < 3$, the samples concentrate near vertices and edges of Δ_3; (b) when $\alpha = 3$, the density is uniform; and (c) when $\alpha > 3$, the density shrinks toward g_0.

most of the probability mass will be contained in a subset of the elements of $\boldsymbol{\pi}$. This phenomenon will be discussed in greater detail in Section 1.4, and is a crucial element of the Dirichlet process. Values of $\boldsymbol{\pi}$ can be drawn from $\text{Dir}(\alpha g_0)$ in a finite number of steps using the following two methods (two infinite-step methods will be discussed shortly).

A Function of Gamma-Distributed Random Variables [78] Gamma-distributed random variables can be used to sample $\boldsymbol{\pi} \sim \text{Dir}(\alpha g_0)$ as follows: Let $Z_i \sim \text{Gamma}(\alpha g_{0i}, \lambda)$ for $i = 1, \ldots, K$, where αg_{0i} is the shape parameter and λ the scale parameter of the gamma distribution. Then the vector $\boldsymbol{\pi} := \left(\frac{Z_1}{\sum_i Z_i}, \ldots, \frac{Z_K}{\sum_i Z_i}\right)$ has a $\text{Dir}(\alpha g_0)$ distribution. The parameter λ can be set to any positive, real value, but must remain constant.

A Function of Beta-Distributed Random Variables[19] Beta-distributed random variables can also be used to draw from the Dirichlet distribution. For $k = 1, \ldots, K-1$,

let

$$\begin{aligned} V_k &\sim \text{Beta}\left(\alpha g_{0k}, \alpha \sum_{\ell=k+1}^{K} g_{0\ell}\right) \\ \pi_k &= V_k \prod_{\ell=1}^{k-1}(1-V_\ell) \\ \pi_K &= 1 - \sum_{k=1}^{K-1} \pi_k \end{aligned} \tag{1.4}$$

The resulting vector, $(\pi_1, \ldots, \pi_K) \sim \text{Dir}(\alpha g_0)$. A second method that uses an infinite number of beta-distributed random variables will be discussed in Section 1.4.

1.2.1 Calculating the Posterior of π

As indicated above, the Dirichlet distribution is conjugate to the multinomial distribution, meaning that the posterior distribution of $\boldsymbol{\pi}$ can be calculated analytically, and is also a Dirichlet distribution. Using Bayes theorem,

$$p(\boldsymbol{\pi}|X=i, \alpha g_0) = \frac{p(X=i|\boldsymbol{\pi})p(\boldsymbol{\pi}|\alpha g_0)}{\int_{\boldsymbol{\pi}\in\Delta_K} p(X=i|\boldsymbol{\pi})p(\boldsymbol{\pi}|,\alpha g_0)d\boldsymbol{\pi}} \propto p(X=i|\boldsymbol{\pi})p(\boldsymbol{\pi}|\alpha g_0)$$

we can calculate the posterior distribution of $\boldsymbol{\pi}$ given the observation X. First, the likelihood of an observation is $p(X=i|\boldsymbol{\pi}) = \pi_i$. Multiplying this by the prior, $p(\boldsymbol{\pi}|\alpha g_0)$, given in (1.2), the posterior is proportional to

$$p(\boldsymbol{\pi}|X=i) \propto \pi_i^{(\alpha g_{0i}+1)-1} \prod_{j\neq i} \pi_j^{\alpha g_{0j}-1} \tag{1.5}$$

The normalizing constant is simply a number that makes this function integrate to one. However, looking at (1.5), it can be seen that this function is proportional to the $\text{Dir}(\alpha g_0 + \mathbf{e}_i)$ distribution. Therefore, the i^{th} parameter of the Dirichlet distribution has simply been incremented by one. This extends naturally to N observations,

$$p(\boldsymbol{\pi}|X_1=x_1,\ldots,X_N=x_N) \propto \prod_{n=1}^{N} p(X_n=x_n|\boldsymbol{\pi})p(\boldsymbol{\pi}|\alpha g_0) = \prod_{i=1}^{K} \pi_i^{\alpha g_{0i}+n_i-1} \tag{1.6}$$

where $n_k = \sum_{n=1}^{N} \mathbf{e}_{X_n}(k)$, or the number of observations taking value k, and $\sum_{k=1}^{K} n_k = N$. When normalized, the posterior is $\text{Dir}(\alpha g_{01} + n_1, \ldots, \alpha g_{0K} + n_K)$. Therefore, when used as a conjugate prior to the multinomial distribution, the posterior of $\boldsymbol{\pi}$ is a Dirichlet distribution, where the parameters have been updated with the "counts" from the observed data.

The interaction between the prior and the data can be seen in the posterior expectation of an element, π_k,

$$\mathbb{E}[\pi_k | X_1 = x_1, \ldots, X_N = x_N] = \frac{n_k + \alpha g_{0k}}{\alpha + N} = \frac{n_k}{\alpha + N} + \frac{\alpha g_{0k}}{\alpha + N} \tag{1.7}$$

The last expression clearly shows the tradeoff between the prior and the data.

A good example of a simple application of these ideas is found at the website IMDB.com.[1] This website allows users to evaluate movies using a star rating system by assigning an integer value between one and ten to a movie, with ten being the best rating. Using these ratings, they provide a ranking of the top 250 movies according to the IMDB community.[2] Clearly they do not want to sort by the empirical average star rating, since this will result in many movies with only a few ratings being highly ranked, and because the average rating of a movie that has many votes can be trusted more than one with only a few votes. They resolve this issue in the following way.

Assume that movie m has an underlying probability vector, $\boldsymbol{\pi}^{(m)} \in \Delta_{10}$, with the probability that a person gives k stars to movie m equal to $\pi_k^{(m)}$. If this vector were known, then the average star rating converges to the expected star rating, which can be calculated analytically and equals $R_m = \sum_{k=1}^{10} k\pi_k^{(m)}$. However, the vector $\boldsymbol{\pi}^{(m)}$ is not known, and so IMDB places a Dirichlet prior on this vector. They set $\alpha = 3000$ and define g_0 to be the empirical distribution of the star values using ratings for *all*

[1] The Internet Movie Database, which at the time of this writing is ranked the 23rd most popular site on the internet in the USA according to Alexa.com.

[2] http://www.imdb.com/chart/top. The definition of their Bayes estimator is given at the bottom of the page and is equivalent to our more detailed description.

movies. They then integrate out $\boldsymbol{\pi}^{(m)}$ and approximate the rating of movie m as

$$\hat{R}_m = \sum_{k=1}^{10} k \frac{n_k}{\alpha + N} + \sum_{k=1}^{10} k \frac{\alpha g_0}{\alpha + N}$$

that is, they use the posterior expectation of $\boldsymbol{\pi}^{(m)}$ given in (1.7) to approximate the rating. The interpretation of α and g_0 is clear in this example: α functions as a prior number of observations, in this case ratings, and αg_{0k} as a prior count of the number of observations in group k. These ratings are "made up" by the person compiling the list. Therefore, the number of ratings actually observed, N, is overwhelmed by this prior at first, but gradually comes to dominate it. As $N \to \infty$, the average rating converges to the average using only the user ratings. This is an example of a Bayes estimate of the value R_m using a Dirichlet prior distribution, and the benefit of its use in this situation is clear.

Returning to the general discussion, two methods requiring an infinite number of random variables, the Pólya urn process and Sethuraman's constructive definition, also exist for sampling $\boldsymbol{\pi} \sim \text{Dir}(\alpha g_0)$. These methods are more complicated in nature, but become essential for sampling from the infinite-dimensional Dirichlet process. Because they are arguably easier to understand in the finite setting (i.e., finite K), they are given here, which will allow for a more intuitive extension to the infinite-dimensional case.

1.3 The Pólya Urn Process

The Pólya urn process [41] is a sequential method for obtaining samples, $X_1, \ldots, X_N$, from a random, discrete probability distribution that has a Dirichlet prior distribution. This process has the following illustrative story: Imagine an urn that initially contains α balls, each of which can take one of K colors; there are αg_{01} balls of the first color, αg_{02} of the second, etc., where $g_0 \in \Delta_K$. A person randomly selects a ball,

X_1, from this urn, with the probability that $X_1 = x_1$ equal to g_{0,x_1}. This person then replaces the ball and adds a second ball of the same color; therefore, the number of balls in the urn increases by one with each draw. This process repeats N times, using the current state of the urn for each sample. After the first draw, the second ball is therefore drawn from the distribution

$$p(X_2|X_1 = x_1) = \frac{1}{\alpha + 1}\delta_{x_1} + \frac{\alpha}{\alpha + 1}\sum_{k=1}^{K} g_{0k}\delta_k$$

where δ_k is a delta function at the color having index k. Inductively, the distribution on the $N + 1^{\text{st}}$ ball is,

$$p(X_{N+1}|X_1 = x_1, \ldots, X_N = x_N) = \sum_{k=1}^{K} \frac{n_k}{\alpha + N}\delta_k + \frac{\alpha}{\alpha + N}\sum_{k=1}^{K} g_{0k}\delta_k \tag{1.8}$$

Comparing with (1.7), these probabilities equal the expectation of $\boldsymbol{\pi}$ under its posterior Dirichlet distribution given $X_1, \ldots, X_N$. Using the rule for integrating out random variables, $\int_{\Omega_B} p(A|B)p(B|C)dB = p(A|C)$, we show this by writing

$$p(X_{N+1}|X_1 = x_1, \ldots, X_N = x_N) = \int_{\boldsymbol{\pi}\in\Delta_K} p(X_{N+1}|\boldsymbol{\pi})p(\boldsymbol{\pi}|X_1 = x_1, \ldots, X_N = x_N)d\boldsymbol{\pi} \tag{1.9}$$

We leave the conditioning on αg_0 as being implied. Since $p(X_{N+1} = i|\boldsymbol{\pi}) = \pi_i$, it follows that for a single value, k, (1.9) is another expression for

$$p(X_{N+1} = k|X_1 = x_1, \ldots, X_N = x_N) = \mathbb{E}[\pi_k|X_1 = x_1, \ldots, X_N = x_N] \tag{1.10}$$

With respect to the Dirichlet distribution and equation (1.9), we are integrating out, or *marginalizing*, the random vector, $\boldsymbol{\pi}$, and are therefore said to be drawing from a marginalized Dirichlet distribution.

Returning to the urn, by the law of large numbers [78], the empirical distribution of the urn converges to some random discrete distribution as the number of samples

$N \to \infty$. The theory of exchangeability [3] shows that the distribution of this random probability mass function is the $\text{Dir}(\alpha g_0)$ distribution.

To review, a sequence of random variables is said to be exchangeable if, for any permutation of the integers $1, \ldots, N$, $\sigma(\cdot)$, it follows that $p(X_1, \ldots, X_N) = p(X_{\sigma(1)}, \ldots, X_{\sigma(N)})$. By sequentially selecting the appropriate values from (1.8) using the chain rule, $p(X_1, \ldots, X_N) = \prod_{i=1}^{N} p(X_i | X_{j<i})$, it can be seen that

$$p(X_1 = x_1, \ldots, X_N = x_N) = \frac{\prod_{k=1}^{K} \prod_{i=1}^{n_k} (\alpha g_{0k} + i - 1)}{\prod_{n=1}^{N} (\alpha + n - 1)} \tag{1.11}$$

where $n_k = \sum_{n=1}^{N} \mathbb{I}(x_n = k)$. This probability does not change for all permutations of the sequence $(X_1, \ldots, X_N)$, and therefore this sequence is exchangeable. As a result of this exchangeability, de Finetti's theorem [28] states that there exists a discrete pmf, $\boldsymbol{\pi}$, having the (yet-to-be-determined) distribution, $p(\boldsymbol{\pi})$, conditioned on which the observations, $(X_1, \ldots, X_N)$, are independent. This is expressed in the following sequence of equalities.

$$\begin{aligned}
p(X_1 = x_1, \ldots, X_N = x_N) &= \int_{\boldsymbol{\pi} \in \Delta_K} p(X_1 = x_1, \ldots, X_N = x_N | \boldsymbol{\pi}) p(\boldsymbol{\pi}) d\boldsymbol{\pi} \\
&= \int_{\boldsymbol{\pi} \in \Delta_K} \prod_{n=1}^{N} p(X_n = x_n | \boldsymbol{\pi}) p(\boldsymbol{\pi}) d\boldsymbol{\pi} \\
&= \int_{\boldsymbol{\pi} \in \Delta_K} \prod_{k=1}^{K} \pi_k^{n_k} p(\boldsymbol{\pi}) d\boldsymbol{\pi} \\
&= \mathbb{E}_p \left[\prod_{k=1}^{K} \pi_k^{n_k} \right]
\end{aligned} \tag{1.12}$$

A distribution is uniquely defined by its moments, and the only distribution having the moments

$$\mathbb{E}_p \left[\prod_{k=1}^{K} \pi_k^{n_k} \right] = \frac{\prod_{k=1}^{K} \prod_{i=1}^{n_k} (\alpha g_{0k} + i - 1)}{\prod_{n=1}^{N} (\alpha + n - 1)} \tag{1.13}$$

is the Dir(αg_0) distribution. The conclusion is that, in the limit as $N \to \infty$, the sequence $(X_1, X_2, \dots)$ drawn according to the urn process defined above can be viewed as independent and identically distributed samples from the pmf $\boldsymbol{\pi}$, where $\boldsymbol{\pi} \sim \text{Dir}(\alpha g_0)$ and is equal to the empirical distribution of the observations.

1.4 Constructing the Finite-Dimensional Dirichlet Distribution

In this section, we review the constructive definition of a finite-dimensional Dirichlet distribution [68]. Like the Pólya urn process, this method also requires an infinite number of random variables to obtain the vector $\boldsymbol{\pi} \sim \text{Dir}(\alpha g_0)$. Unlike this process, the values of the observations $X_1, \dots, X_N$ are not required to do this, but rather random variables are drawn *iid* from their respective distributions, and $\boldsymbol{\pi}$ is constructed according to a function of these random variables. The details of the proof is given in this section because this will be the central process from which the novel stick-breaking construction of the beta process is derived in Chapter 5.

The constructive definition of a Dirichlet prior states that, if $\underline{\pi}$ is constructed according to the following function of random variables, then $\underline{\pi} \sim \text{Dir}(\alpha g_0)$.

$$\begin{aligned} \underline{\pi} &= \sum_{i=1}^{\infty} V_i \prod_{j=1}^{i-1} (1 - V_j) \underline{\mathrm{e}}_{Y_i} \\ V_i &\overset{iid}{\sim} \text{Beta}(1, \alpha) \\ Y_i &\overset{iid}{\sim} \text{Mult}(\{1, \dots, K\}, g_0) \end{aligned} \tag{1.14}$$

Using this multinomial parametrization, $Y \in \{1, \dots, K\}$ with $\mathbb{P}(Y = k | g_0) = g_{0k}$. The vector $\underline{\mathrm{e}}_Y$ is a K-dimensional vector of zeros, except for a one in position Y. The values $V_i \prod_{j=1}^{i-1}(1 - V_j)$ are often called "stick-breaking" weights, because at step i, the proportion V_i is "broken" from the remainder, $\prod_{j=1}^{i-1}(1 - V_j)$, of a unit-length stick. Since $V \in [0, 1]$, the product $V_i \prod_{j=1}^{i-1}(1 - V_j) \in [0, 1]$ for all i, and it can be

shown that $\sum_{i=1}^{\infty} V_i \prod_{j=1}^{i-1}(1-V_j) = 1$.

In Figure 1.2, we illustrate this process for $i = 1, \ldots, 4$. The weights are broken as mentioned, and the random variables $\{Y_i\}_{i=1}^{4}$ indicate the elements of the vector $\underline{\pi}$ to which each weight is added. In the limit, the value of the k^{th} element is

$$\pi_k = \sum_{i=1}^{\infty} V_i \prod_{j=1}^{i-1}(1-V_j)\mathbb{I}(Y_i = k)$$

where $\mathbb{I}(\cdot)$ is the indicator function and equals one when the argument is true, and zero otherwise.

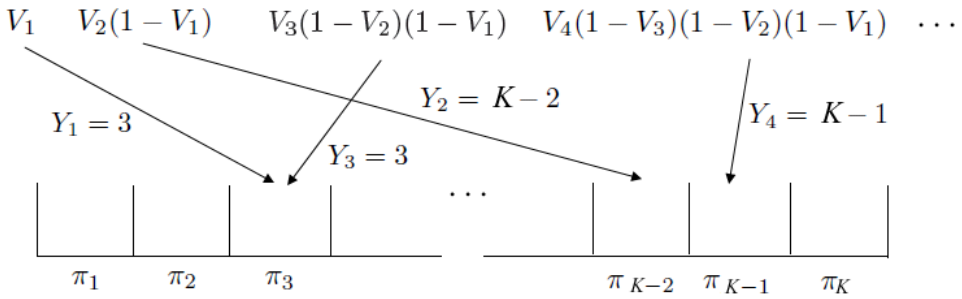


FIGURE 1.2: An illustration of the infinite stick-breaking construction of a K-dimensional Dirichlet distribution. Weights are drawn according to a Beta(1,α) stick-breaking process, with corresponding locations taking value k with probability g_{0k}.

1.4.1 Proof of the Construction

We begin with the random vector

$$\underline{\pi} \sim \text{Dir}(\alpha g_0) \tag{1.15}$$

The proof that this random vector has the same distribution as the random vector in (1.14) requires two lemmas concerning general properties of the Dirichlet distribution, the first of which follows.

Lemma 1: Let $Z \sim \sum_{k=1}^{K} g_{0k}\text{Dir}(\alpha g_0 + \underline{e}_k)$. Values of Z can be sampled from this distribution by first sampling $Y \sim \text{Mult}(\{1, \ldots, K\}, g_0)$, and then sampling $Z \sim \text{Dir}(\alpha g_0 + \underline{e}_Y)$. It then follows that $Z \sim \text{Dir}(\alpha g_0)$.

The proof of this lemma is in the appendix at the end of this chapter. Therefore, the process

$$\begin{aligned} \underline{\pi} &\sim \text{Dir}(\alpha g_0 + \underline{e}_Y) \\ Y &\sim \text{Mult}(\{1, \ldots, K\}, g_0) \end{aligned} \tag{1.16}$$

produces a random vector $\underline{\pi} \sim \text{Dir}(\alpha g_0)$. The second lemma, which will be applied to the result in (1.16), is

Lemma 2: Let the random vectors $W_1 \sim \text{Dir}(w_1, \ldots, w_K)$, $W_2 \sim \text{Dir}(v_1, \ldots, v_K)$ and $V \sim \text{Beta}(\sum_{k=1}^{K} w_k, \sum_{k=1}^{K} v_k)$. Define the linear combination,

$$Z := VW_1 + (1 - V)W_2 \tag{1.17}$$

then $Z \sim \text{Dir}(w_1 + v_1, \ldots, w_K + v_K)$.

The proof of this lemma is in the appendix at the end of this chapter. In words, this lemma states that, if one wished to construct the vector $Z \in \Delta_K$ according to the function of random variables (W_1, W_2, V) given in (1.17), one could equivalently bypass this construction and directly sample $Z \sim \text{Dir}(w_1 + v_1, \ldots, w_K + v_K)$.

This lemma is applied to the random vector $\underline{\pi} \sim \text{Dir}(\alpha g_0 + \underline{e}_Y)$ in (1.16), with

the result that this vector can be represented by the following process,

$$\begin{aligned}
\underline{\pi} &= VW + (1-V)\underline{\pi}' \\
W &\sim \text{Dir}(\underline{\text{e}}_Y) \\
\underline{\pi}' &\sim \text{Dir}(\alpha g_0) \\
V &\sim \text{Beta}\left(\sum_{k=1}^{K} \underline{\text{e}}_Y(k), \sum_{k=1}^{K} \alpha g_{0k}\right) \\
Y &\sim \text{Mult}(\{1,\ldots,K\}, g_0)
\end{aligned} \tag{1.18}$$

where we've also included the random variable Y. The result is still a random vector $\underline{\pi} \sim \text{Dir}(\alpha g_0)$. Note that $\sum_{k=1}^{K} \underline{\text{e}}_Y(k) = 1$ and $\sum_{k=1}^{K} \alpha g_{0k} = \alpha$. Also, we observe that the distribution of W is degenerate, with only one of the K parameters in the Dirichlet distribution being nonzero. Therefore, since $\mathbb{P}(\pi_k = 0 | g_{0k} = 0) = 1$, we can say that $\mathbb{P}(W = \underline{\text{e}}_Y | g_0 = \underline{\text{e}}_Y) = 1$. Modifying (1.18), the following generative process for $\underline{\pi}$ produces a random vector $\underline{\pi} \sim \text{Dir}(\alpha g_0)$.

$$\begin{aligned}
\underline{\pi} &= V\underline{\text{e}}_Y + (1-V)\underline{\pi}' \\
\underline{\pi}' &\sim \text{Dir}(\alpha g_0) \\
V &\sim \text{Beta}\,(1, \alpha) \\
Y &\sim \text{Mult}(\{1,\ldots,K\}, g_0)
\end{aligned} \tag{1.19}$$

We observe that the random vectors $\underline{\pi}$ and $\underline{\pi}'$ have the same distribution, which is a desired result. That is, returning to (1.14), we expand the right hand term as follows

$$\begin{aligned}
\underline{\pi} &= V_1 \underline{\text{e}}_{Y_1} + (1-V_1) \sum_{i=2}^{\infty} V_i \prod_{j=2}^{i-1} (1-V_j) \underline{\text{e}}_{Y_i} \\
&= V_1 \underline{\text{e}}_{Y_1} + (1-V_1)\underline{\pi}'
\end{aligned} \tag{1.20}$$

Due to the iid nature of the random variables $\{V_i\}_{i=1}^{\infty}$ and $\{Y_i\}_{i=1}^{\infty}$, the vector $\underline{\pi}$ has

the same distribution as the vector $\underline{\pi}'$ in (1.20), which we have shown is the $\text{Dir}(\alpha g_0)$ distribution.

Since $\underline{\pi}' \sim \text{Dir}(\alpha g_0)$ in (1.19), this vector can be decomposed according to the same process by which $\underline{\pi}$ is decomposed in (1.19). Thus, for $i = 1, 2$ we have

$$\begin{aligned} \underline{\pi} &= V_1 \underline{\text{e}}_{Y_1} + V_2(1 - V_1)\underline{\text{e}}_{Y_2} + (1 - V_1)(1 - V_2)\underline{\pi}'' \\ V_i &\overset{iid}{\sim} \text{Beta}(1, \alpha) \\ Y_i &\overset{iid}{\sim} \text{Mult}(\{1, \ldots, K\}, g_0) \\ \underline{\pi}'' &\sim \text{Dir}(\alpha g_0) \end{aligned} \tag{1.21}$$

which can proceed for $i \to \infty$, with each decomposition leaving the vector $\underline{\pi} \sim \text{Dir}(\alpha g_0)$. In the limit as $i \to \infty$, (1.21)→(1.14), since $\lim_{i\to\infty} \prod_{j=1}^{i}(1 - V_j) = 0$, concluding the proof. Therefore, (1.14) arises as an infinite number of decompositions of the Dirichlet distribution, each taking the form of (1.19).

1.5 The Extension to Infinite-Dimensional Spaces

In this section we briefly make the connection between the stick-breaking representation of the finite Dirichlet distribution and the infinite Dirichlet process. The extension of the Pólya urn process to infinite-dimensional spaces [11], also called the Chinese restaurant process [3], follows a similar argument. For finite-dimensional vectors, $\underline{\pi} \in \Delta_K$, the constructive definition of (1.14) may seem unnecessary, since the infinite sum cannot be carried out in practice, and $\underline{\pi}$ can be constructed exactly using only K gamma-distributed random variables. The primary use of the stick-breaking representation of the Dirichlet distribution is the case where $K \to \infty$.

For example, consider a K-component mixture model [37], where observations in

a data set, $\{X_n\}_{n=1}^N$, are generated according to $X_n \sim p(\theta_n^*)$ and $\theta_n^* \overset{iid}{\sim} G_K$, where

$$\begin{aligned} G_K &= \sum_{k=1}^{K} \pi_k \delta_{\theta_k} \\ \underline{\pi} &\sim \text{Dir}(\alpha g_0) \\ \theta_k &\overset{iid}{\sim} G_0, \quad k = 1, \ldots, K \end{aligned} \tag{1.22}$$

The atom θ_n^* associated with observation X_n contains parameters for some distribution, $p(\cdot)$, with $\mathbb{P}(\theta_n^* = \theta_k | \underline{\pi}) = \pi_k$. The $\text{Dir}(\alpha g_0)$ prior is often placed on $\underline{\pi}$ as shown, and G_0 is a (typically non-atomic) base distribution. For the Gaussian mixture model [26], $\theta_k = \{\mu_k, \Sigma_k\}$ and G_0 is often a conjugate Normal-Wishart prior distribution on mean vector μ and covariance matrix Σ.

Following the proof of Section 1.4.1, we can let (1.14) construct $\underline{\pi}$ in (1.22), producing

$$\begin{aligned} G_K &= \sum_{i=1}^{\infty} V_i \prod_{j=1}^{i-1} (1 - V_j) \delta_{\theta_{Y_i}} \\ V_i &\overset{iid}{\sim} \text{Beta}(1, \alpha) \\ Y_i &\overset{iid}{\sim} \text{Mult}\left(\{1, \ldots, K\}, g_0\right) \\ \theta_k &\overset{iid}{\sim} G_0, \quad k = 1, \ldots, K \end{aligned} \tag{1.23}$$

Sampling Y_i from the integers $\{1, \ldots, K\}$ according to g_0 provides an index of the atom with which to associate mass $V_i \prod_{j=1}^{i-1}(1 - V_j)$. Ishwaran and Zarepour [37] have shown that, when $g_0 = (\frac{1}{K}, \ldots, \frac{1}{K})$ and $K \to \infty$, $G_K \to G$, where G is a Dirichlet process with continuous base measure G_0 on the infinite space $(S, \mathcal{A})$, as defined in [27]; this definition is given at the end of this section. Since in the limit as $K \to \infty$, $\mathbb{P}(Y_i = Y_j | i \neq j) = 0$ and $\mathbb{P}(\theta_{Y_i} = \theta_{Y_j} | i \neq j) = 0$, there is a one-to-one correspondence between $\{Y_i\}_{i=1}^{\infty}$ and $\{\theta_i\}_{i=1}^{\infty}$. Let the function $\sigma(Y_i) = i$ reindex the subscripts on

θ_{Y_i}. Then

$$\begin{aligned} G &= \sum_{i=1}^{\infty} V_i \prod_{j=1}^{i-1}(1-V_j)\delta_{\theta_i} \\ V_i &\overset{iid}{\sim} \text{Beta}(1,\alpha) \\ \theta_i &\overset{iid}{\sim} G_0 \end{aligned} \tag{1.24}$$

Finally, we note that this representation is not as different from (1.14) as first appears. For example, let G_0 be a *discrete* measure on the first K positive integers. In this case $\theta \in \{1,\ldots,K\}$ and (1.24) and (1.14) are essentially the same (the difference being that one is presented as a measure on the first K integers, while the other is a vector in Δ_K).

As mentioned, we provide a more formal definition of a Dirichlet process first given in [27] here. Let S be an abstract space and let $\mu(\cdot)$ be a measure on that space with the measure of the entire space $\mu(S) = \alpha$. For any subset A of S, let $G_0(A) = \mu(A)/\mu(S)$. Then G is a Dirichlet procees if for all partitions of S, $A_1,\ldots,A_k$,

$$(G(A_1),\ldots,G(A_k)) \sim \text{Dir}(\alpha G_0(A_1),\ldots,\alpha G_0(A_k))$$

In the machine learning community, G_0 is almost always non-atomic and can be thought of as corresponding to a continuous probability distribution. The value of $G_0(A)$ is simply the integration of the corresponding density over the region A. Therefore, a Dirichlet process, G, produces a probability measure on the subsets $A_1,\ldots,A_k$ of the space S by drawing from a Dirichlet distribution with concentration parameter α and $g_0 := (G_0(A_1),\ldots,G_0(A_k))$.

1.6 Inference for the Dirichlet Process Mixture Model

The discussion thus far has led to methods for generating $G \sim \mathrm{DP}(\alpha G_0)$. In this section, we discuss the use of G, as well as Markov chain Monte Carlo (MCMC) inference [29] for mixture models using the stick-breaking representation of G.

1.6.1 Dirichlet Process Mixture Models

The primary use of the Dirichlet process in the machine learning community is in nonparametric mixture modeling [5][26], where values $\theta \sim G$ are not the observed data, as X was in the previous sections, but rather a parameter or set of parameters for some density function, $f_X(\theta)$, from which X is drawn. As discussed in the previous section, the generative process is,

$$\begin{aligned} X_n &\sim f_X(\theta_n^*) \\ \theta_n^* &\overset{iid}{\sim} G \\ G &\sim \mathrm{DP}(\alpha G_0) \end{aligned} \tag{1.25}$$

and the parameter θ_n^* is a specific value selected from $G = \sum_{i=1}^{\infty} p_i \delta_{\theta_i}$ and is associated with observation X_n. While samples drawn from G will contain duplicate values, samples from $f_X(\theta)$ are again from a continuous density. Therefore, values that share parameters are *clustered* together in that, though not exactly the same, they exhibit similar statistical characteristics according to the density function, $f_X(\theta)$.

1.6.2 Gibbs Sampling for Dirichlet Process Mixture Models

We next outline a general method for performing MCMC [29] inference for Dirichlet process mixture models. The MCMC engine presented here is the Gibbs sampler [4], which to review, samples values for each parameter in a sequential fashion using the posterior distribution of a parameter of interest conditioned on the current values of

all other parameters. We focus on inference using the stick-breaking construction of the DP, which is fully conjugate and therefore amenable to Gibbs sampling inference.

Let $f_X(x|\theta)$ be the likelihood function of an observation, x, given parameters, θ, and let $p(\theta)$ be the prior density of θ. Inference for Dirichlet process mixture models also includes an additional latent variable [26], $c \in \{1, \ldots, K\}$, which indicates the parameter value associated with observation X_n. We also define a K-truncated Dirichlet process (to be discussed in greater detail in the next chapter) to be the process in (1.24) where $i = 1, \ldots, K$ and the remaining probability mass is added to the last atom. We define the K-dimensional vector $\boldsymbol{p} = \phi_K(\boldsymbol{V})$ to be the resulting probability weights. Incorporating c and using the truncated DP, the generative process is,

$$\begin{aligned} X_n &\sim f_X(\theta_{c_n}) \\ c_n &\overset{iid}{\sim} \text{Mult}(\{1, \ldots, K\}, \phi_K(\boldsymbol{V})) \\ V_k &\overset{iid}{\sim} \text{Beta}(1, \alpha) \\ \theta_k &\overset{iid}{\sim} G_0 \end{aligned} \tag{1.26}$$

The MCMC sampling procedure is given below.

Initialization: Select a truncation level, K, and initialize the model, G, by sampling $\theta_k \sim G_0$ for $k = 1, \ldots, K$ and $V_k \sim \text{Beta}(1, \alpha)$ for $k = 1, \ldots, K-1$ and construct $\boldsymbol{p} = \phi_K(\boldsymbol{V})$. We discuss a method for learning α at the end of the section, but it is assumed to be initialized here.

Step 1: Sample the indicators, $c_1, \ldots, c_N$, independently from their respective con-

ditional posteriors, $p(c_n|x_n, G) \propto f_X(x_n|\theta_{c_n})p(\theta_{c_n}|G)$,

$$c_n \sim \sum_{k=1}^{K} \frac{p_k f_X(x_n|\theta_k)}{\sum_l p_l f_X(x_n|\theta_l)} \delta_k \tag{1.27}$$

Relabel the index values to remove any unused integers between one and $\max c_n$. Let the total number of unique values be denoted by K'. Set K to be some integer value larger than K'. These additional components act as proposals that allow for the data to use more components if required.

Step 2: Sample $\theta_1, \ldots, \theta_{K'}$ from their respective posteriors conditioned on $c_1, \ldots, c_N$ and $x_1, \ldots, x_N$,

$$\begin{aligned} \theta_k &\sim p\left(\theta_k | \{c_n\}_{n=1}^N, \{x_n\}_{n=1}^N\right) \\ p\left(\theta_k | \{c_n\}_{n=1}^N, \{x_n\}_{n=1}^N\right) &\propto \prod_{n=1}^{N} f_X(x_n|\theta)^{\mathbb{I}(c_n=k)} p(\theta) \end{aligned} \tag{1.28}$$

the function $\mathbb{I}(c_n = k)$ is used to pick out which x_n belong to component k. Sample $\theta_{K'+1}, \ldots, \theta_K \overset{iid}{\sim} G_0$.

Step 3: For the stick-breaking process, construct the K-dimensional weight vector, $\boldsymbol{p} = \phi_K(\boldsymbol{V})$, using $V_1, \ldots, V_{K-1}$ sampled from their beta-distributed posteriors conditioned on $c_1, \ldots, c_N$,

$$V_k \sim \text{Beta}\left(1 + \sum_{n=1}^{N} \mathbb{I}(c_n = k), \alpha + \sum_{\ell > k} \sum_{n=1}^{N} \mathbb{I}(c_n = \ell)\right) \tag{1.29}$$

Set $p_K = \prod_{k=1}^{K-1}(1 - V_k)$. For the Chinese restaurant process, $K := K' + 1$ and each p_k is replaced by $p_k = \frac{n_k}{\alpha+N}$ for $k = 1, \ldots, K-1$, and $p_K = \frac{\alpha}{\alpha+N}$.

Repeat Steps 1 – 3 for a desired number of iterations. The convergence of this Markov chain can be assessed [29], after which point samples from these steps adequately spaced out in the chain are considered iid samples from the full posterior, $p\left(\{\theta_k\}_{k=1}^K, \{V_k\}_{k=1}^K, \{c_n\}_{n=1}^N, K|\{x_n\}_{n=1}^N\right)$. Additional inference for α can be performed for the CRP using a method detailed in [26], and for the stick-breaking construction using a conjugate gamma prior.

Step 4: Sample α from its posterior gamma distribution conditioned on $V_1, \ldots, V_{K'}$

$$\alpha \sim \text{Gamma}\left(a + K', b - \sum_{k=1}^{K'} \ln(1 - V_k)\right) \tag{1.30}$$

where a, b are the parameters for the gamma prior distribution.

As can be seen, inference for the Dirichlet process is fairly straightforward and, when $p(\theta)$ is conjugate to $f_X(x|\theta)$, it is fully analytical. Other MCMC methods exist for performing this inference [49] as does a fast, variational inference method [13][8] that converges deterministically to a local optimal approximation to the full posterior distribution following initialization. The objective function that this variational method attempts to minimize is the Kullback-Liebler divergence between the approximation of the full posterior and true joint posterior of all parameters.

1.7 The Expectation and Variance of the Entropy of Dirichlet Processes

In this section, we derive the expectation and variance of the entropy of discrete measures sampled from Dirichlet distributions and, by extension to the infinite limit, the Dirichlet process with non-atomic base measure G_0. To our knowledge, this calculation is a new contribution to the theory of Dirichlet processes.

The entropy of $\underline{\pi} \sim \mathrm{Dir}(\alpha g_0)$, denoted $\mathrm{H}(\underline{\pi})$, is

$$\mathrm{H}(\underline{\pi}) = -\sum_{k=1}^{K} \pi_k \ln \pi_k \tag{1.31}$$

The expectation to be calculated is,

$$\mathbb{E}[\mathrm{H}(\underline{\pi})|\alpha g_0] = \int_{\Delta_K} \left(-\sum_{k=1}^{K} \pi_k \ln \pi_k \right) p(\underline{\pi}|\alpha g_0)\ d\underline{\pi} \tag{1.32}$$

The variance, $\mathbb{V}[\mathrm{H}(\underline{\pi})|\alpha g_0] = \mathbb{E}[\mathrm{H}(\underline{\pi})^2|\alpha g_0] - \mathbb{E}[\mathrm{H}(\underline{\pi})|\alpha g_0]^2$, is

$$\mathbb{V}[\mathrm{H}(\underline{\pi})|\alpha g_0] = \int_{\Delta_K} \left(\sum_{i=1}^{K} \sum_{j=1}^{K} \pi_i \pi_j \ln \pi_i \ln \pi_j \right) p(\underline{\pi}|\alpha g_0)\ d\underline{\pi} - \mathbb{E}[\mathrm{H}(\underline{\pi})|\alpha g_0]^2 \tag{1.33}$$

We give these two values below for a finite Dirichlet distribution, and the infinite Dirichlet process (where $g_0 = (1/K, \ldots, 1/K)$, $K \to \infty$), denoted $G \sim \mathrm{DP}(\alpha G_0)$, followed by their derivations.

$$\mathbb{E}[\mathrm{H}(\underline{\pi})|\alpha g_0] = \psi(\alpha+1) - \sum_{k=1}^{K} g_{0k}\psi(\alpha g_{0k}+1) \tag{1.34}$$

$$\begin{aligned}\mathbb{V}[\mathrm{H}(\underline{\pi})|\alpha g_0] = & \\ & \sum_{k=1}^{K} \frac{(\alpha g_{0k}+1)g_{0k}}{\alpha+1}\left[(\psi(\alpha g_{0k}+2)-\psi(\alpha+2))^2 + \psi'(\alpha g_{0k}+2) - \psi'(\alpha+2)\right] \\ + & \sum_{i\neq j} \frac{\alpha g_{0i} g_{0j}}{\alpha+1}\left[(\psi(\alpha g_{0i}+1)-\psi(\alpha+2))(\psi(\alpha g_{0j}+1)-\psi(\alpha+2)) - \psi'(\alpha+2)\right] \\ - & \left(\psi(\alpha+1) - \sum_{k=1}^{K} g_{0k}\psi(\alpha g_{0k}+1)\right)^2\end{aligned} \tag{1.35}$$

$$\mathbb{E}[\mathrm{H}(G)|\alpha G_0] = \psi(\alpha+1) - \psi(1) \tag{1.36}$$

$$\begin{aligned}\mathbb{V}[\mathrm{H}(G)|\alpha G_0] = & \frac{1}{\alpha+1}(\psi(2)-\psi(\alpha+2))^2 + \frac{\alpha}{\alpha+1}(\psi(1)-\psi(\alpha+2))^2 \\ & + \frac{\psi'(2)}{\alpha+1} - \psi'(\alpha+2) - (\psi(\alpha+1)-\psi(1))^2\end{aligned} \tag{1.37}$$

The function $\psi(\cdot)$ is the digamma function and $\psi'(\cdot)$ is the first derivative of the digamma function. In Figure 1.3, we plot $\mathbb{E}[\mathrm{H}(G)|\alpha G_0]$ vs $\mathbb{V}[\mathrm{H}(G)|\alpha G_0]$ for the values $\alpha = 0.1, 0.2, 0.3, \ldots, 1000$. We see that, for the Dirichlet process, as α increases, the variance around the expected entropy decreases.

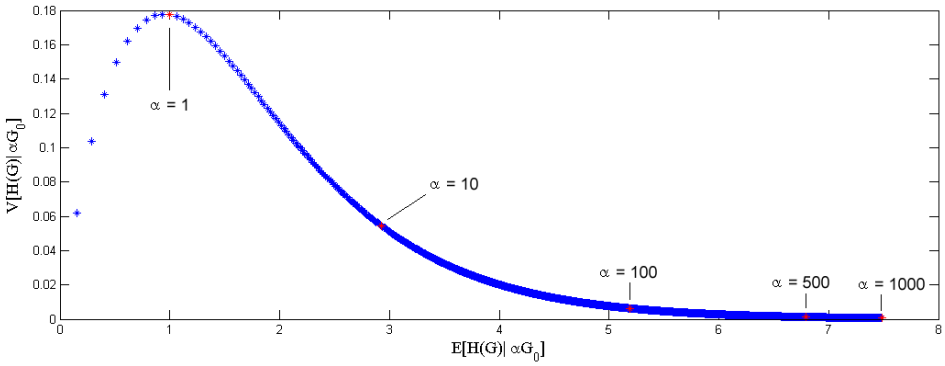


FIGURE 1.3: The expectation and variance of the entropy of $G \sim \mathrm{DP}(\alpha G_0)$ for α between 0.1 and 1000 with steps of 0.1. As can be seen, the expected entropy increases as α increases, while the variance of this entropy decreases for $\alpha > 1$.

1.7.1 Derivation of the Expectation

The sequence of equalities is presented first, after which a discussion shows how one follows another. Let e_k be a K-dimensional vector of zeros, except for a one in the k^{th} dimension.

$$\begin{aligned}
\mathbb{E}[\mathrm{H}(\underline{\pi})|\alpha g_0] &= \int_{\Delta_K} \left(-\sum_{k=1}^{K} \pi_k \ln \pi_k \right) p(\underline{\pi}|\alpha g_0)\ d\underline{\pi} && (1.38)\\
&= -\sum_{k=1}^{K} \int_{\Delta_K} (\pi_k \ln \pi_k) p(\underline{\pi}|\alpha g_0)\ d\underline{\pi} && (1.39)\\
&= -\sum_{k=1}^{K} \frac{\Gamma(\alpha)\Gamma(\alpha g_{0k}+1)}{\Gamma(\alpha g_{0k})\Gamma(\alpha+1)} \int_{\Delta_K} (\ln \pi_k) p(\underline{\pi}|\alpha g_0 + \mathrm{e}_k)\ d\underline{\pi} && (1.40)\\
&= -\sum_{k=1}^{K} g_{0k} \mathbb{E}[\ln \pi_k | \alpha g_0 + \mathrm{e}_k] && (1.41)\\
&= -\sum_{k=1}^{K} g_{0k} (\psi(\alpha g_{0k}+1) - \psi(\alpha+1)) && (1.42)\\
&= \psi(\alpha+1) - \sum_{k=1}^{K} g_{0k} \psi(\alpha g_{0k}+1) && (1.43)
\end{aligned}$$

Note that $\Gamma(n+1) = n\Gamma(n)$. Also, when $g_0 = (1/K, \ldots, 1/K)$,[3]

$$\mathbb{E}[\mathrm{H}(G)|\alpha G_0] = \lim_{K\to\infty} \mathbb{E}[\mathrm{H}(\underline{\pi})|\alpha g_0] \tag{1.44}$$

by monotone convergence. It therefore follows from the above derivation that

$$\mathbb{E}[\mathrm{H}(G)|\alpha G_0] = \psi(\alpha+1) - \psi(1) \tag{1.45}$$

To go from (1.39) to (1.40) the π_k is absorbed in the Dirichlet distribution and the trick of multiplying and dividing by the same thing is used to turn this into a $\mathrm{Dir}(\alpha g_0 + \mathrm{e}_k)$ distribution, which is where the coefficients come from. These coefficients simplify as shown in (1.41), and the expectation of $\ln \pi_k$ is a familiar calculation from variational inference. Things then simplify further as shown until (1.43). For the infinite DP, all $g_{0k} = 1/K$, which allows the last term of (1.43) to come outside the sum, which then sums to one. Letting $K \to \infty$ produces (1.45).

1.7.2 Derivation of the Variance

We want to calculate

$$\begin{aligned}
\mathbb{V}[\mathrm{H}(\underline{\pi})|\alpha g_0] &= \int_{\Delta_K} \left(\sum_{i=1}^{K} \sum_{j=1}^{K} \pi_i \pi_j \ln \pi_i \ln \pi_j \right) p(\underline{\pi}|\alpha g_0)\ d\underline{\pi} - \mathbb{E}[\mathrm{H}(\underline{\pi})|\alpha g_0]^2 \\
&= \sum_{i=1}^{K} \sum_{j=1}^{K} \int_{\Delta_K} (\pi_i \pi_j \ln \pi_i \ln \pi_j) p(\underline{\pi}|\alpha g_0)\ d\underline{\pi} - \mathbb{E}[\mathrm{H}(\underline{\pi})|\alpha g_0]^2
\end{aligned} \tag{1.46}$$

[3] As previously discussed, Ishwaran and Zarepour [37] have shown that, for a uniform g_0, when the elements of $\underline{\pi} \sim \mathrm{Dir}(\alpha g_0)$ are used as probability masses on atoms drawn *iid* from a non-atomic base probability measure, G_0, then as $K \to \infty$, this process converges to $G \sim \mathrm{DP}(\alpha G_0)$.

The two important terms are $\mathbb{E}[\pi_i \pi_j \ln \pi_i \ln \pi_j | \alpha g_0]$ for $i \neq j$ and $\mathbb{E}[(\pi_i \ln \pi_i)^2 | \alpha g_0]$, which we derive below. First using the same reasoning as in Section 1.7.1,

$$\begin{aligned}
\mathbb{E}[(\pi_i \ln \pi_i)^2 | \alpha g_0] &= \frac{\Gamma(\alpha)\Gamma(\alpha g_{0i} + 2)}{\Gamma(\alpha + 2)\Gamma(\alpha g_{0i})} \mathbb{E}[(\ln \pi_i)^2 | \alpha g_0 + 2\mathrm{e}_i] \\
&= \frac{(\alpha g_{0i} + 1) g_{0i}}{\alpha + 1} \mathbb{E}[(\ln \pi_i)^2 | \alpha g_0 + 2\mathrm{e}_i] \qquad (1.47)
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}[\pi_i \pi_j \ln \pi_i \ln \pi_j | \alpha g_0] &= \frac{\Gamma(\alpha)\Gamma(\alpha g_{0i} + 1)\Gamma(\alpha g_{0j} + 1)}{\Gamma(\alpha + 2)\Gamma(\alpha g_{0i})\Gamma(\alpha g_{0j})} \mathbb{E}[\ln \pi_i \ln \pi_j | \alpha g_0 + \mathrm{e}_i + \mathrm{e}_j] \\
&= \frac{\alpha g_{0i} g_{0j}}{\alpha + 1} \mathbb{E}[\ln \pi_i \ln \pi_j | \alpha g_0 + \mathrm{e}_i + \mathrm{e}_j] \qquad (1.48)
\end{aligned}$$

Calculate $\mathbb{E}[(\ln \boldsymbol{\pi_i})^2 | \boldsymbol{\alpha g_0 + 2e_i}]$: We know from variational inference that

$$\mathbb{E}[\ln \pi_i | \alpha g_0 + 2\mathrm{e}_i] = \psi(\alpha g_{0i} + 2) - \psi(\alpha + 2) \qquad (1.49)$$

Taking the partial derivative of each side with respect to αg_{0i} produces

$$\int_{\Delta_K} (\ln \pi_i)(\psi(\alpha+2) - \psi(\alpha g_{0i}+2) + \ln \pi_i) p(\underline{\pi} | \alpha g_0 + 2\mathrm{e}_i)\, d\underline{\pi} = \psi'(\alpha g_{0i}+2) - \psi'(\alpha+2) \qquad (1.50)$$

and therefore

$$\mathbb{E}[(\ln \pi_i)^2 | \alpha g_0 + 2\mathrm{e}_i] = (\psi(\alpha g_{0i} + 2) - \psi(\alpha + 2))^2 + \psi'(\alpha g_{0i} + 2) - \psi'(\alpha + 2) \qquad (1.51)$$

Calculate $\mathbb{E}[\ln \boldsymbol{\pi_i} \ln \boldsymbol{\pi_j} | \boldsymbol{\alpha g_0 + e_i + e_j}]$: Start with the equality

$$\mathbb{E}[\ln \pi_i | \alpha g_0 + \mathrm{e}_i + \mathrm{e}_j] = \psi(\alpha g_{0i} + 1) - \psi(\alpha + 2) \qquad (1.52)$$

and take the partial derivative of each side with respect to αg_{0j},

$$\int_{\Delta_K} (\ln \pi_i)(\psi(\alpha+2) - \psi(\alpha g_{0j}+1) + \ln \pi_j) p(\underline{\pi} | \alpha g_0 + \mathrm{e}_i + \mathrm{e}_j)\, d\underline{\pi} = -\psi'(\alpha+2) \qquad (1.53)$$

This simplifies to

$$\mathbb{E}[\ln \pi_i \ln \pi_j | \alpha g_0 + \mathrm{e}_i + \mathrm{e}_j] = (\psi(\alpha g_{0i} + 1) - \psi(\alpha + 2))(\psi(\alpha g_{0j} + 1) - \psi(\alpha + 2)) - \psi'(\alpha + 2) \tag{1.54}$$

Inserting these two values into (1.46) and using the result for $\mathbb{E}[\mathrm{H}(\underline{\pi})|\alpha g_0]$ in Section 1.7.1 produces

$$\begin{aligned}
\mathbb{V}[\mathrm{H}(\underline{\pi})|\alpha g_0] = & \\
& \sum_{k=1}^{K} \frac{(\alpha g_{0k} + 1) g_{0k}}{\alpha + 1} \left[(\psi(\alpha g_{0k} + 2) - \psi(\alpha + 2))^2 + \psi'(\alpha g_{0k} + 2) - \psi'(\alpha + 2) \right] \\
+ & \sum_{i \neq j} \frac{\alpha g_{0i} g_{0j}}{\alpha + 1} \left[(\psi(\alpha g_{0i} + 1) - \psi(\alpha + 2)) (\psi(\alpha g_{0j} + 1) - \psi(\alpha + 2)) - \psi'(\alpha + 2) \right] \\
- & \left(\psi(\alpha + 1) - \sum_{k=1}^{K} g_{0k} \psi(\alpha g_{0k} + 1) \right)^2
\end{aligned} \tag{1.55}$$

which is the value given in (1.35).

As with the expected entropy, we insert $g_0 = (1/K, \ldots, 1/K)$ and let $K \to \infty$ to obtain the corresponding result for the Dirichlet process using the monotone convergence theorem,

$$\begin{aligned}
\mathbb{V}[\mathrm{H}(G)|\alpha G_0] &= \frac{1}{\alpha + 1} (\psi(2) - \psi(\alpha + 2))^2 + \frac{\alpha}{\alpha + 1} (\psi(1) - \psi(\alpha + 2))^2 \\
&\quad + \frac{\psi'(2)}{\alpha + 1} - \psi'(\alpha + 2) - (\psi(\alpha + 1) - \psi(1))^2
\end{aligned} \tag{1.56}$$

1.8 Appendix

Proof of Lemma 1: Let $Y \sim \text{Mult}(\{1, \ldots, K\}, \underline{\pi})$ and $\underline{\pi} \sim \text{Dir}(\alpha g_0)$. Basic probability theory allows us to write that

$$p(\underline{\pi}|\alpha g_0) = \sum_{k=1}^{K} \mathbb{P}(Y = k|\alpha g_0) p(\underline{\pi}|Y = k, \alpha g_0) \tag{1.57}$$

$$\mathbb{P}(Y = k|\alpha g_0) = \int_{\underline{\pi} \in \Delta_K} \mathbb{P}(Y = k|\underline{\pi}) p(\underline{\pi}|\alpha g_0)\, d\underline{\pi} \tag{1.58}$$

The second equation can be written as $\mathbb{P}(Y = k|\alpha g_0) = \mathbb{E}[\pi_k|\alpha g_0] = g_{0k}$. The first equation uses the posterior of a Dirichlet distribution given observation $Y = k$, which is $p(\underline{\pi}|Y = k, \alpha g_0) = \text{Dir}(\alpha g_0 + \mathbf{e}_k)$. Replacing these two equalities in the first equation, we obtain $p(\underline{\pi}|\alpha g_0) = \sum_{k=1}^{K} g_{0k} \text{Dir}(\alpha g_0 + \mathbf{e}_k)$.

Proof of Lemma 2: We use the representation of $\underline{\pi}$ as a function of gamma-distributed random variables. That is, if $\gamma_k \sim \text{Gamma}(\alpha g_{0k}, \lambda)$ for $k = 1, \ldots, K$, and we define $\underline{\pi} := \left(\frac{\gamma_1}{\sum_k \gamma_k}, \ldots, \frac{\gamma_K}{\sum_k \gamma_k}\right)$, then $\underline{\pi} \sim \text{Dir}(\alpha g_0)$. Using this definition, let

$$W_1 := \left(\frac{\gamma_1}{\sum_k \gamma_k}, \ldots, \frac{\gamma_K}{\sum_k \gamma_k}\right),\ W_2 := \left(\frac{\gamma'_1}{\sum_k \gamma'_k}, \ldots, \frac{\gamma'_K}{\sum_k \gamma'_k}\right),\ V := \frac{\sum_k \gamma_k}{\sum_k \gamma_k + \sum_k \gamma'_k} \tag{1.59}$$

where $\gamma_k \sim \text{Gamma}(w_k, \lambda)$ and $\gamma'_k \sim \text{Gamma}(v_k, \lambda)$. Then it follows that

$$W_1 \sim \text{Dir}(w_1, \ldots, w_K),\ W_2 \sim \text{Dir}(v_1, \ldots, v_K),\ V \sim \text{Beta}(\textstyle\sum_k w_k, \sum_k v_k) \tag{1.60}$$

where the distribution of V arises because $\sum_k \gamma_k \sim \text{Gamma}(\sum_k w_k, \lambda)$. Furthermore, Basu's theorem [7] indicates that V is independent of W_1 and W_2, or $p(V|W_1, W_2) = p(V)$. Performing the multiplication $Z = VW_1 + (1 - V)W_2$ produces the gamma-distributed representation of $Z \sim \text{Dir}(w_1 + v_1, \ldots, w_K + v_K)$.

2

Sethuraman's Constructive Definition for Finite Mixture Models

2.1 Abstract

In this chapter, we investigate Sethuraman's construction of the finite-dimensional Dirichlet distribution [68], specifically when used as a prior for mixture modeling. In Section 2.2, we consider two error properties of the finite-dimensional Dirichlet distribution with reference to the infinite Dirichlet process. In Section 2.3, we consider two applications of the infinite stick-breaking representation of the finite Dirichlet prior, (i) as a means for performing additional inference for α using a conjugate gamma prior, and (ii) as a means for performing fully conjugate inference for the hierarchical Dirichlet process.

2.2 Comparing $\mathrm{Dir}(\alpha g_0)$ and $\mathrm{DP}(\alpha G_0)$ Priors Using Constructive Definitions

The stick-breaking representation of the finite-dimensional Dirichlet distribution provides additional insight into the similar performance of this prior to the infinite-dimensional prior for mixture modeling. That is, using finite-symmetric Dirichlet

priors to perform mixture modeling, (1.22) tends in practice to be as "nonparametric" as the infinite-dimensional representation in (1.23) when K is large. In this section, we use the representation in (1.14) to investigate how $\underline{\pi} \sim \mathrm{Dir}(\alpha g_0)$ relates to $G \sim \mathrm{DP}(\alpha G_0)$ as priors for mixture models; see [36][37][38] for further analysis. Since $G_K \to G$ when $g_0 = (\frac{1}{K}, \ldots, \frac{1}{K})$ as discussed in Section 1.5, we assume g_0 is uniform in what follows.

Consider the remaining length of a unit-length stick after the i^{th} break from a $\mathrm{Beta}(1, \alpha)$ stick-breaking process,

$$\epsilon := \prod_{j=1}^{i} (1 - V_j)$$

In terms of a truncated approximation to the DP, where the stick-breaking process in (1.24) is terminated after the K^{th} break and ϵ is arbitrarily assigned to a set of (possibly one) atoms, this value can also be viewed as the error of the approximation [36][48]. As K increases, this value decreases in expectation, with $\mathbb{E}[\epsilon|\alpha, K] = \left(\frac{\alpha}{1+\alpha}\right)^K$.

2.2.1 Statistical Properties of ϵ

More precise statistical properties of this error can also be derived. For example, define the random variable $S_j := -\ln(1 - V_j)$. It then can be shown that $S_j \sim \mathrm{Exponential}(\alpha)$, and therefore $-\ln \epsilon = \sum_{j=1}^{i} S_j \sim \mathrm{Gamma}(i, \alpha)$. This gives the generative process for ϵ,

$$\begin{aligned} \epsilon &= \mathrm{e}^{-T} \\ T &\sim \mathrm{Gamma}(i, \alpha) \end{aligned} \tag{2.1}$$

which, using a change of variable, $T = f(\epsilon) = -\ln \epsilon$, produces the density

$$p(\epsilon|\alpha, i) = \frac{\alpha^i}{(i-1)!} \epsilon^{\alpha-1} (\ln 1/\epsilon)^{i-1} \tag{2.2}$$

Likewise, consider when the value of ϵ is fixed and let the positive integer-valued random variable $L = \arg\min_L \prod_{j=1}^{L}(1 - V_j) < \epsilon$ [48]. Using the definitions of S_j and T above, this can equivalently be written as $L = \arg\min_L \sum_{j=1}^{L} S_j > T$. Since each S_j has an exponential distribution, the value of $L - 1$ is a Poisson process on $\mathbb{R}_+$ with stopping time T and exponential waiting times having rate α [43]. Therefore, $L^- := L - 1 \sim \text{Poisson}(\alpha T)$, and thus a generative process for L is

$$\begin{aligned} L &= L^- + 1 \\ L^- &\sim \text{Poisson}(-\alpha \ln \epsilon) \end{aligned} \tag{2.3}$$

which, for a $\text{Beta}(1, \alpha)$ stick-breaking process, is the distribution on the number of breaks of the unit-length stick before the remaining length is smaller than ϵ.

These two representations of ϵ and L can be used to characterize how many breaks will be made before the error to the DP is small. For example, when $\alpha = 2$ and $\epsilon = 0.01$, Poisson(9.21) breaks will be made prior to truncation, with $\mathbb{E}[L|\alpha = 2, \epsilon = 0.01] = 10.21$.

2.2.2 Connecting $\text{Dir}(\alpha g_0)$ *and* $\text{DP}(\alpha G_0)$ *Priors Via* ϵ

These same statistical properties of the error apply to the finite-dimensional Dirichlet distribution, as is evident in (1.14), with the exception being that the error is now with respect to the construction of the vector $\underline{\pi} \in \Delta_K$. Therefore, when $L \ll K$, the resulting vector, $\underline{\pi}$, will be sparse, since *at most* only L of the K components will contain mass defined as meaningful (i.e., prior to truncation). If we sample

$\underline{\pi} \sim \text{Dir}(\alpha g_0)$ *exactly*, then the decomposition in (1.21) allows us to write this as

$$\begin{aligned} \underline{\pi} &= \sum_{i=1}^{L} V_i \prod_{j=1}^{i-1}(1-V_j)\underline{e}_{Y_i} + \epsilon\underline{\pi}' \\ V_i &\overset{iid}{\sim} \text{Beta}(1,\alpha) \\ Y_i &\overset{iid}{\sim} \text{Mult}(\{1,\ldots,K\}, g_0) \\ \underline{\pi}' &\sim \text{Dir}(\alpha g_0) \end{aligned} \tag{2.4}$$

Therefore, splitting the error, ϵ, according to the fractions $\underline{\pi}' \sim \text{Dir}(\alpha g_0)$ produces a vector $\underline{\pi} \sim \text{Dir}(\alpha g_0)$ with at most L components containing mass defined as meaningful.

The number of components containing the first L masses can clearly be smaller than L, since $\mathbb{P}(Y_i = Y_j | i \neq j, i \leq L, j \leq L) > 0$. However, when this number is equal to L an equivalence occurs between the finite-dimensional Dirichlet distribution and the L-truncated Dirichlet process in the mixture modeling framework. For example, inserting (2.4) into (1.22) yields

$$\begin{aligned} G_K &= \sum_{i=1}^{L} V_i \prod_{j=1}^{i-1}(1-V_j)\delta_{\theta_{Y_i}} + \sum_{k=1}^{K} \epsilon\pi'_k \delta_{\theta_k} \\ V_i &\overset{iid}{\sim} \text{Beta}(1,\alpha) \\ Y_i &\overset{iid}{\sim} \text{Mult}\left(\{1,\ldots,K\}, g_0\right) \\ \underline{\pi}' &\sim \text{Dir}(\alpha g_0) \\ \theta_k &\overset{iid}{\sim} G_0, \quad k = 1,\ldots,K \end{aligned} \tag{2.5}$$

In what follows, we will refer to $\sum_{i=1}^{L} V_i \prod_{j=1}^{i-1}(1-V_j)\delta_{\theta_{Y_i}}$ the "significant measure" and $\sum_{k=1}^{K} \epsilon\pi'_k\delta_{\theta_k}$ as the "residual measure." The sum of these measures produces the

finite Dirichlet-distributed measure on the atoms $\{\theta_1, \ldots, \theta_K\}$ given in (1.22); G_K is still a K-dimensional mixture with a $\text{Dir}(\alpha g_0)$ prior on the mixing weights.

Similarly, the L-truncation of the infinite Dirichlet process, G_{tr}^L, can be written as

$$\begin{aligned} G_{\text{tr}}^L &= \sum_{i=1}^{L} V_i \prod_{j=1}^{i-1}(1-V_j)\delta_{\theta_i} + \sum_{\ell=1}^{\infty} \epsilon_\ell \delta_{\theta_\ell} \\ V_i &\overset{iid}{\sim} \text{Beta}(1, \alpha) \\ \theta_i &\overset{iid}{\sim} G_0 \end{aligned} \tag{2.6}$$

where $\sum_{\ell=1}^{\infty} \epsilon_\ell = \epsilon = \prod_{i=1}^{L}(1-V_j)$, with each ϵ_ℓ defined arbitrarily. For example, assigning ϵ to the last component before truncation means that $\epsilon_L = \epsilon$ and $\epsilon_\ell = 0$ for $\ell \neq L$ [36]. Again, the significant and residual measures are $\sum_{i=1}^{L} V_i \prod_{j=1}^{i-1}(1-V_j)\delta_{\theta_i}$ and $\sum_{\ell=1}^{\infty} \epsilon_\ell \delta_{\theta_\ell}$, respectively; the significant measure in the first line of (2.6) is what defines G_{tr}^L as an L-truncated Dirichlet process.

The representation of G_K in (2.5), and G_{tr}^L in (2.6) are only distinguishable by their significant measures when $\epsilon_\ell := 0$ for $\ell > K$ and $(\epsilon_1/\epsilon, \ldots, \epsilon_K/\epsilon) \sim \text{Dir}(\alpha g_0)$ (a modification of [36]), and ignoring indexing of the exchangeable atoms. Specifically, using the representation in (2.5), a G_K measure can be distinguished from a G_{tr}^L measure by observing that two weights, e.g., $V_i \prod_{j=1}^{i-1}(1-V_j)$ and $V_i' \prod_{j=1}^{i'-1}(1-V_j)$, are assigned to the same atom, θ_{Y_i} and $\theta_{Y_{i'}}$, in cases where $Y_i = Y_{i'}$. When $\sum_{i \neq j} \mathbb{I}(Y_i = Y_j) = 0$ for $i \leq L, j \leq L$, G_K is indistinguishable from G_{tr}^L, i.e., G_K is an L-truncated Dirichlet process. The probability of this event is equal to $\mathbb{P}(\sum_{i \neq j} \mathbb{I}(Y_i = Y_j) = 0 | g_0, i \leq L, j \leq L)$. When g_0 is uniform,

$$\mathbb{P}\left(\sum_{i \neq j} \mathbb{I}(Y_i = Y_j) = 0 | g_0 = (1/K, \ldots, 1/K), i \leq L, j \leq L\right) = \prod_{m=0}^{L-1} \frac{K-m}{K} \tag{2.7}$$

Viewed as an urn process, this is the probability that L draws can be made uniformly

with replacement from an urn containing K balls without drawing the same ball twice.

As can be seen from (2.7), this probability equals zero when $L > K$, which is expected, since L weights are being placed on K atoms. As K increases, this probability increases. As $K \to \infty$, this probability converges to one for all values of L; an infinite Dirichlet process always contains an L-truncated Dirichlet process within its stick-breaking representation. Therefore, Sethuraman's construction of the finite-dimensional Dirichlet prior not only can be used to show why $\text{Dir}(\alpha g_0)$ priors are sparse when $g_0 = \left(\frac{1}{K}, \ldots, \frac{1}{K}\right)$ and $\alpha \ll K$, but it also shows that, using a $\text{Dir}(\alpha g_0)$ prior on the weights of a K-dimensional mixture model, G_K, there is a probability (2.7) that G_K will be equivalent to an L-truncated draw from $\text{DP}(\alpha G_0)$.

Furthermore, using the definition of error given in [36] as being the assignment of probability mass to atoms that have previously been drawn (specifically, the last atom), we can define this same error, ϵ_{DD}, for the K-dimensional Dirichlet distribution with uniform g_0. In Figure 2.1, we show an illustration of this definition of error. For the Dirichlet process, the K-truncated Dirichlet process and the K-dimensional Dirichlet distribution, the stick-breaking constructions are identical. However, the K-truncated Dirichlet process selects the first K masses and locations from the Dirichlet process, but violates the definition of the Dirichlet process by assigning the sum of masses indexed by $K+1, K+2, \ldots$ to a previously observed atom; according to the Dirichlet process, these masses should be assigned to new atoms drawn *iid* from G_0. For sake of argument, we can take the K locations of the finite Dirichlet distribution to correspond to those of the Dirichlet process. However, the policy of assigning the stick-breaking weights of the finite Dirichlet distribution can violate that of the Dirichlet process *before* the first K weights have been assigned. Any time a mass is added to a previously observed atom, this contradicts the Dirichlet process. We call this mass an error to the DP and sum these masses to obtain ϵ_{DD}.

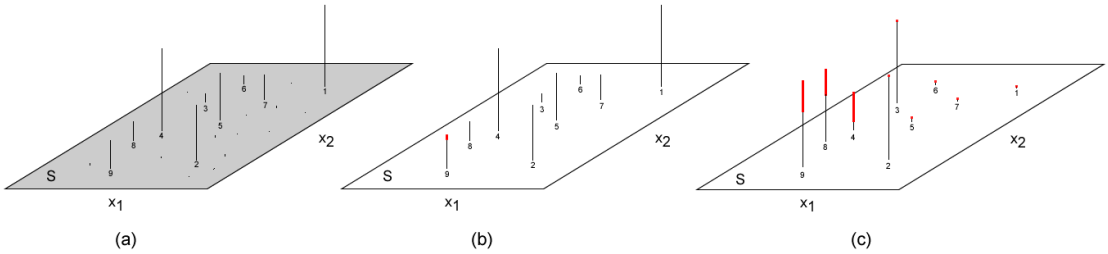


FIGURE 2.1: An illustration of error to the Dirichlet process of the truncated Dirichlet process and finite-dimensional Dirichlet distribution mixture model for 9 mixture components. All priors are defined over the two-dimensional space $(S, \mathcal{A}, G_0)$. The verticle lines indicate probability weights at a given location. All three priors share the first 9 locations and a single stick-breaking process. Shown are (a) The Dirichlet process, where gray indicates that all locations are included in the construction. (b) The $K = 9$ truncation of the Dirichlet process. The remaining mass, ϵ, is added *in error* to the 9^{th} atom, indicated by red. (c) The finite Dirichlet distribution. Because the index of the location of each mass is drawn *iid* from g_0, sticks can be placed on top of other sticks at any point following the first break. All sticks for which this occurs are colored red, and these sticks are distributed in violation (i.e., in error) of the policy for Dirichlet processes.

Below, we give the expectation of ϵ_{DD},

$$\mathbb{E}[\epsilon_{\text{DD}}|\alpha, K] = \frac{\alpha}{\alpha + K} \tag{2.8}$$

The derivation is given in the appendix of this chapter. Using this definition of ϵ_{DD}, the expected error to the Dirichlet process using a $\text{Dir}(\alpha g_0)$ prior with g_0 uniform decreases like the inverse of the dimensionality, K, of the Dirichlet distribution. Not surprisingly, the expected probability mass that is assigned to previously-observed locations is larger for the finite-dimensional Dirichlet distribution than it is for the K-truncation of the infinite DP, where we recall that $\mathbb{E}[\epsilon|\alpha, K] = \left(\frac{\alpha}{1+\alpha}\right)^K$. This is because the first K draws constituting a K-truncated DP – the K largest masses in expectation – have unique locations with probability one, which, as mentioned above, is not the case in (1.23).

2.3 Applications of the Construction of $\text{Dir}(\alpha g_0)$

We consider two applications of the constructive representation of the Dirichlet distribution given in (1.14): (i) As a means for performing inference for α in mixture models with $\text{Dir}(\alpha g_0)$ priors, and (ii) as a means for performing inference for the hierarchical Dirichlet process [70]. The first application is specifically for finite-dimensional mixture models using finite Dirichlet priors. The second application is for infinite-dimensional mixture models where the base measure is discrete, but countably infinite. However, a truncation of the top-level DP renders all second-level DP's finite as well. An advantage of conjugate inference is that Gibbs sampling can be used [29]. Contrary to other sampling methods, such as Metropolis-Hastings [29], where proposed samples can be rejected and a proposal distribution can be difficult to design, Gibbs sampling provides a new sample from the true conditional posterior distribution with each iteration [4]. This can lead to faster mixing of the Markov chain to the stationary distribution. We consider synthetic examples for each proposed sampling method.

2.3.1 Inference for α Using the Constructive Definition of $\text{Dir}(\alpha g_0)$

Consider the fully data-generative process for a finite-dimensional mixture model with a Dirichlet prior. For data $\{X_n\}_{n=1}^N$, this is

$$\begin{aligned} X_n &\sim f_X(\theta_{c_n}) \\ c_n &\overset{iid}{\sim} \text{Mult}(\{1,\ldots,K\},\underline{\pi}) \\ \underline{\pi} &\sim \text{Dir}(\alpha g_0) \\ \theta_k &\overset{iid}{\sim} G_0, \quad k=1,\ldots,K \end{aligned} \tag{2.9}$$

The set of latent indicator variables $\{c_n\}_{n=1}^N$, where $c \in \{1,\ldots,K\}$, select parameters θ_{c_n} for the density function $f_X(\cdot)$ from which observation X_n is generated [26]. In this

section, we show how the stick-breaking representation of the Dirichlet distribution can be used to perform additional inference on α. Though this representation is not required to infer this parameter, it has the advantage of allowing the use of a conjugate gamma prior on α [13], which results in an analytically tractable inference procedure.

First, let the probability vector $\omega_i \sim \text{Dir}(\epsilon g_0)$ and let $\epsilon \to 0$. Then from the stick-breaking construction of this Dirichlet distribution (1.14), it is clear that ω_i is a vector of zeros except for a single one, and has the same distribution as $\underline{e}_{Y_i}$, where $Y_i \sim \text{Mult}(\{1, \ldots, K\}, g_0)$. Therefore, using this random variable and the stick-breaking construction of $\underline{\pi} \sim \text{Dir}(\alpha g_0)$, the generative process of (2.9) can be rewritten as

$$\begin{aligned}
X_n &\sim f_X(\theta_{c_n}) \\
c_n &\sim \sum_{k=1}^{K} \omega_{d_n,k} \delta_k \\
d_n &\overset{iid}{\sim} \sum_{i=1}^{\infty} V_i \prod_{j=1}^{i-1} (1 - V_j) \delta_i \\
V_i &\overset{iid}{\sim} \text{Beta}(1, \alpha) \\
\omega_i &\overset{iid}{\sim} \text{Dir}(\epsilon g_0), \quad \epsilon \to 0 \\
\theta_k &\overset{iid}{\sim} G_0, \quad k = 1, \ldots, K
\end{aligned} \tag{2.10}$$

A second collection of latent indicator variables, $\{d_n\}_{n=1}^N$, has been introduced in this representation. We also observe that c_n is deterministic given d_n, since ω_{d_n} is a delta measure on the integers $\{1, \ldots, K\}$.

We propose the following procedure for learning α: (i) Given α, perform inference for $\{c_n\}_{n=1}^N$ using a finite representation of $\text{Dir}(\alpha g_0)$ – the stick-breaking representation is unnecessary; (ii) Conditioned on $\{c_n\}_{n=1}^N$, sample $\{d_n\}_{n=1}^N$ using collapsed

inference, where the random variables $\{V_i\}_{i=1}^{\infty}$ and $\{\omega_i\}_{i=1}^{\infty}$ are integrated out; (iii) Given $\{d_n\}_{n=1}^{N}$, sample $\{V_i\}_{i=1}^{\max d_n}$ independently from their posterior beta distributions; (iv) Sample α from its posterior gamma distribution conditioned on $\{V_i\}_{i=1}^{\max d_n}$. We detail these steps below.

Step (i), the sampling of $\{c_n\}_{n=1}^{N}$, is a standard procedure for mixture modeling with finite Dirichlet priors (e.g., see [9]), and so we do not review this step here. We have also reviewed this idea in step 1 of Section 1.6.2, though for a stick-breaking prior. For step (ii), the sampling of $\{d_n\}_{n=1}^{N}$ takes the form

$$\begin{aligned} p(d_m = h | c_m, \{V_i\}_{i=1}^{h}, \omega_h) &\propto p(c_m|\omega_h) p(d_m = h | \{V_i\}_{i=1}^{h}) \\ &\propto \omega_{h,c_m} V_h \prod_{j=1}^{h-1} (1 - V_j) \end{aligned} \tag{2.11}$$

However, as mentioned, we do not know the parameters ω_h and $\{V_i\}_{i=1}^{h}$ that render the $\{d_n\}_{n=1}^{N}$ conditionally independent. We therefore integrate them out as follows

$$\begin{aligned} & p\left(d_m = h | c_m, \{c_n\}_{n=1}^{m-1}, \{d_n\}_{n=1}^{m-1}, \alpha g_0\right) \\ & \propto p\left(c_m | d_m = h, \{d_n\}_{n=1}^{m-1}, \{c_n\}_{n=1}^{m-1}, g_0\right) p\left(d_m = h | \{d_n\}_{n=1}^{m-1}, \alpha\right) \\ & \propto \int_{\Delta_K} p\left(c_m|\omega_h\right) p\left(\omega_h | \{d_n\}_{n=1}^{m-1}, \{c_n\}_{n=1}^{m-1}\right) d\omega_h \ldots \\ & \quad \cdots \times \int_{[0,1]^h} p\left(d_m = h | \{V_i\}_{i=1}^{h}\right) p\left(\{V_i\}_{i=1}^{h} | \{d_n\}_{n=1}^{m-1}, \alpha\right) dV_1 \ldots dV_h \end{aligned} \tag{2.12}$$

There are two terms of interest to compute, plus a normalizing constant, each of which is discussed below.

The Likelihood term $p\left(c_m|\cdot\right) = \int_{\Delta_K} p\left(c_m|\cdot\right) p\left(\omega_h|\cdot\right) d\omega_h$: In this term, draws are being made from a marginal Dirichlet distribution where the weights, ω_h, have been

integrated out. The result is a Pólya urn process [11][3] for the h^{th} urn having K initial balls distributed as g_0, and where the total mass of these balls, ϵ, is vanishing. Therefore, for the first draw from a given urn, h, the integer-value of the resulting indicator, c, is distributed as g_0. Any remaining draws from this urn are from a delta function located at this same integer.

More specifically, for this urn process, the indicator c_m is drawn from the probability measure

$$c_m|d_m = h \;\sim\; \sum_{k=1}^{K} \frac{\sum_{n=1}^{m-1} \mathbb{I}(c_n = k)\mathbb{I}(d_n = h) + \epsilon g_{0k}}{\sum_{n=1}^{m-1} \mathbb{I}(d_n = h) + \epsilon}\; \delta_k \tag{2.13}$$

Two cases arise when $\epsilon \to 0$: (i) If c_m is the first observation to come from component h (where h indexes components for the stick-breaking construction of the finite Dirichlet distribution, not for the mixing components), then it is equal to k with probability g_{0k}; (ii) If c_m is not the first draw from component h, as indicated by $\sum_{n=1}^{m-1} \mathbb{I}(d_n = h) > 0$, then the prior vanishes and all subsequent samples are equal to the first value from component h with probability one.

The Prior term $p\left(d_m = h|\cdot\right) = \int_{[0,1]^h} p\left(d_m = h|\cdot\right) p\left(\{V_i\}_{i=1}^h|\cdot\right) dV_1 \ldots dV_h$: For this prior, there are h beta-distributed random variables to be integrated out, with

$$V_j \sim \text{Beta}\left(1 + \sum_{n=1}^{m-1} \mathbb{I}(d_n = j), \alpha + \sum_{n=1}^{m-1} \sum_{\ell > j} \mathbb{I}(d_n = \ell)\right)$$

Given that $p\left(d_m = h|\{V_i\}_{i=1}^h\right) = V_h \prod_{j=1}^{h-1}(1 - V_j)$, the marginalized probability is

$$p\left(d_m = h|\{d_n\}_{n=1}^{m-1}, \alpha\right) = \tag{2.14}$$

$$\frac{1 + \sum_{n=1}^{m-1} \mathbb{I}(d_n = h)}{1 + \alpha + \sum_{n=1}^{m-1} \sum_{\ell \geq h} \mathbb{I}(d_n = \ell)} \prod_{j=1}^{h-1} \frac{\alpha + \sum_{n=1}^{m-1} \sum_{\ell > j} \mathbb{I}(d_n = \ell)}{1 + \alpha + \sum_{n=1}^{m-1} \sum_{\ell \geq j} \mathbb{I}(d_n = \ell)}$$

This prior has a corresponding illustrative restaurant story, which we call the *Chinese restaurant district*. A customer walks down a street containing an infinite number of Chinese restaurants. Upon reaching each restaurant entrance, he chooses to either enter the restaurant, or continue walking down the street. He enters restaurant h with a probability nearly proportional to the number of customers in the restaurant, and continues walking down the street with a probability nearly proportional to the number of customers in restaurants still to be encountered. A second way to view this process is as a marginalization of a coin flipping process, where coins having different biases are sequentially flipped until a head is observed. The probability of observing a head from the j^{th} coin is equal to V_j. The value in (2.14) is the marginal probability of observing the first head from coin h given all previous d_n.

The Normalizing constant: The final step is to calculate the normalizing constant for the distribution on the latent indicator d_m,

$$Z_m = \sum_{h=1}^{\infty} p\left(c_m | d_m = h, \{d_n\}_{n=1}^{m-1}, \{c_n\}_{n=1}^{m-1}, g_0\right) p\left(d_m = h | \{d_n\}_{n=1}^{m-1}, \alpha\right)$$

Let $d_{\max}^{(m)} = \max_{n<m} d_n$ and define $\eta_m := \prod_{j=1}^{d_{\max}^{(m)}} \frac{\alpha + \sum_{n=1}^{m-1} \sum_{\ell > j} \mathbb{I}(d_n = \ell)}{1 + \alpha + \sum_{n=1}^{m-1} \sum_{\ell \geq j} \mathbb{I}(d_n = \ell)}$. Then it follows that

$$Z_m = \sum_{h=1}^{d_{\max}^{(m)}} p\left(c_m | d_m = h, \{d_n\}_{n=1}^{m-1}, \{c_n\}_{n=1}^{m-1}, g_0\right) p\left(d_m = h | \{d_n\}_{n=1}^{m-1}, \alpha\right) + \eta_m g_{0,c_m} \tag{2.15}$$

where $p\left(c_m | d_m = h, \{d_n\}_{n=1}^{m-1}, \{c_n\}_{n=1}^{m-1}, g_0\right)$ and $p\left(d_m = h | \{d_n\}_{n=1}^{m-1}, \alpha\right)$ are from (2.13) and (2.14), respectively.

Given $\{d_n\}_{n=1}^{N}$, the random variables $V_1, \ldots, V_{d_{\max}^{(N)}}$ can be sampled independently

from their respective posterior distributions,

$$V_j \sim \text{Beta}\left(1 + \sum_{n=1}^{N} \mathbb{I}(d_n = j), \alpha + \sum_{n=1}^{N} \sum_{\ell > j} \mathbb{I}(d_n = \ell)\right) \tag{2.16}$$

following which, α can be sampled from its posterior gamma distribution using a conjugate Gamma(a, b) prior,

$$\alpha \sim \text{Gamma}\left(a + d_{\max}^{(N)}, b - \sum_{j=1}^{d_{\max}^{(N)}} \ln(1 - V_j)\right) \tag{2.17}$$

Given this new α, inference can again proceed for $\{c_n\}_{n=1}^{N}$ using the finite representation of the Dirichlet distribution. Furthermore, if MAP or variational inference are performed [9], in which case we do not have an integer value for c_n, but rather a distribution on values for c_n, these values can be sampled according to their current distributions prior to executing the procedure outlined in this section.

In Figure 2.2, we show results for a synthetic example. Values of α were sampled from a Uniform$(1, 25)$ distribution, followed by the sampling of $\underline{\pi} \sim \text{Dir}(\alpha g_0)$, where $g_0 \in \Delta_{50}$ and uniform. This was followed by sampling $c_n \sim \text{Mult}(\{1, \ldots, 50\}, \underline{\pi})$ for $n = 1, \ldots, 1000$. The above inference method was run for 20 iterations with an initial value of $\alpha = 1$. The sample of the 20^{th} iteration was then taken to be the inferred value for α, shown on the y-axis.

As is evident from the figure, there is an estimation bias above the true value, as marked by the red line. While the slope of this line is equal to one, the slope of the line fitted to the data using least squares equals 1.3537. Observing the inference method in practice, this appears to be due to the following reason: The number of V_j sampled according to (2.16) which are then used to sample α according to (2.17) is equal to the maximum value of $\{d_n\}_{n=1}^{N}$. It is often the case that some of these d_n will be "outliers" in the sense that they are very large and there are many values,

$j < d_{\max}^{(N)}$, for which no $d_n = j$. However, since there is data in the corresponding posterior of V_j, these sampled V_j (of which there are a large number) are still used to infer α. The small values of these V_j encourage the posterior of α to inflate by increasing the posterior mean. This causes a larger value of α to be sampled for the next iteration, which encourages more outliers. We've observed that these outliers tend to come for lower indexed values of d_n. In an attempts to mitigate this phenomenon, we considered only including those V_j for which $d_n = j$ for some n. This, however, resulted in significant underestimation of α; the data tended to fall below the red line. Other ad hoc heuristics also did not resolve this issue.

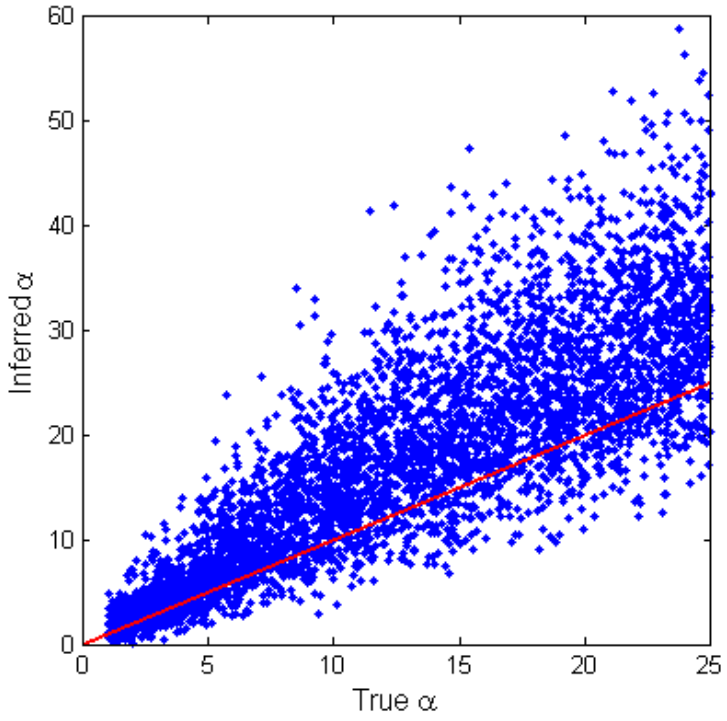


FIGURE 2.2: Using Sethuraman's construction to infer the value of α in the $\text{Dir}(\alpha g_0)$ distribution. A total of 5000 trials are shown using synthetic data. In each trial, a true value for α was randomly generated, followed by a vector $\underline{\pi} \sim \text{Dir}(\alpha g_0)$ and $N = 1000$ samples from $\underline{\pi}$. Each point in the plot indicates the inferred value of α compared with the actual value.

2.3.2 Inference for the Hierarchical Dirichlet Process

The procedure for sampling α given in the previous section can be modified and expanded to perform inference for the hierarchical Dirichlet process (HDP). To review, the HDP [70] constitutes draws from a Dirichlet process for which the base measure is itself drawn from a Dirichlet process. We focus on two levels in this section,

$$\begin{aligned} G &\sim \mathrm{DP}(\alpha G_0) \\ G'_m &\overset{iid}{\sim} \mathrm{DP}(\beta G) \end{aligned} \tag{2.18}$$

The base distribution for the second-level Dirichlet process, G, is discrete with probability one [68], which means that each draw, G'_m for $m = 1, \ldots, M$, represents a mixture for which the atoms are shared, but the weights on these atoms differ for each m. This is seen clearly in the following equivalent representation.

$$\begin{aligned} G = \sum_{i=1}^{\infty} w_i \delta_{\theta_i}, \qquad & w_i := V_i \prod_{j=1}^{i-1} (1 - V_j) \\ V_i &\overset{iid}{\sim} \mathrm{Beta}(1, \alpha) \\ \theta_i &\overset{iid}{\sim} G_0 \\ (G'_m(\theta_1), G'_m(\theta_2), \ldots) &\overset{iid}{\sim} \mathrm{Dir}(\beta w_1, \beta w_2, \ldots) \end{aligned} \tag{2.19}$$

The top-level DP is constructed according to the stick-breaking construction given in (1.24), from which it follows that each G'_m is a draw from an infinite-dimensional Dirichlet distribution with $g_0 := (w_1, w_2, \ldots)$ no longer being uniform. The hierarchical Dirichlet process is useful for document modeling [70] and multitask learning [18] where multiple documents or tasks are expected to share a collection of distributions (i.e., components), but with each group using these components differently according to their own probabilities.

We can use Sethuraman's stick-breaking construction to represent G'_m as well,

$$\begin{aligned} G'_m &= \sum_{i=1}^{\infty} V'_{m,i} \prod_{j=1}^{i-1} (1 - V'_{m,j}) \delta_{\theta_{Y_{m,i}}} \\ V'_{m,i} &\overset{iid}{\sim} \text{Beta}(1, \beta) \\ Y_{m,i} &\overset{iid}{\sim} \sum_{k=1}^{\infty} w_k \delta_k \end{aligned} \tag{2.20}$$

We note that this representation is not the stick-breaking representation given in [70] for the second-level DP, which is instead equivalent to the representation in (1.4). Using this representation and the ideas in Section 2.3.1, inference can be performed for α, β and $V_1, \ldots, V_{\text{K}}$ of the top-level DP, where K will be the maximum value of a set of latent indicators similar to that in the previous section; all priors will again be conjugate, allowing for analytical posterior calculations, and all other parameters will be integrated out to facilitate the calculation of these posteriors.

Let $\{X_n^{(m)}\}_{n=1}^{N_m}$ be the set of observations from the m^{th} task to be modeled by the second-level DP, G'_m. For the first step, we assume that values for α, β and $V_1, \ldots, V_K$ have been drawn from their conditional posteriors of the previous iteration. This allows for the construction of $w_1, \ldots, w_K$, using $V_1, \ldots, V_K$, as well as further values of $w_{K+1}, w_{K+2}, \ldots$, to be constructed via draws from the prior, $V_{K+h} \sim \text{Beta}(1, \alpha)$. In this chapter, we assume that the top-level DP is truncated at some level, $K_t \geq K$, which yields a finite, K_t-dimensional Dirichlet distribution prior for each second-level task. Let $g_0 := (w_1, \ldots, w_{K_t})$. Therefore, g_0 is changing in both its values and its size with each iteration, but for any given iteration it is the same across each of the M tasks.

The fully generative process for tasks $m = 1, \ldots, M$ is then of the form given in (2.10), with the exception being that the unknowns X_n, c_n and d_n have a superscript of m, indicating the task to which they belong, and V_i is replaced with $V'_{m,i}$. Thus,

we again introduce a second set of latent indicators, $\{d_n^{(m)}\}_{n=1}^{N_m}$, which serve the same purpose as in Section 2.3.1 and can be obtained using the same method. The only difference is that α in the previous section is now β, since α here is the scaling parameter of the top-level DP. The posterior of β therefore follows from the previous section and is

$$\beta \sim \text{Gamma}\left(a + \sum_{m=1}^{M} d_{\max}^{(N_m)}, b - \sum_{m=1}^{M} \sum_{j=1}^{d_{\max}^{(N_m)}} \ln(1 - V'_{m,j})\right) \tag{2.21}$$

where $d_{\max}^{(N_m)} = \max_n d_n^{(m)}$.

It remains to calculate the posteriors of α and $V_1, \ldots, V_K$. We observe in the last line of (2.20) that, if these values of $Y_{m,i}$ were available, posteriors for $V_1, V_2, \ldots$ could be calculated. In fact, several of these values are implicitly available in the sets $\{c_n^{(m)}\}_{n=1}^{N_m}$ and $\{d_n^{(m)}\}_{n=1}^{N_m}$, and we propose to find and use these values. The value $Y_{m,i}$ indicates that component i of the stick-breaking representation of the second-level DP for the m^{th} task is a delta function on component $Y_{m,i}$ of this DP (we again note the distinction between components of the second-level DP, and components of the stick-breaking representation of this DP). Therefore, if $d_n^{(m)} = i$, it follows that $Y_{m,i} = c_n^{(m)}$. It's clear from the discussion of the previous section that not all values of i will be represented in the set $\{d_n^{(m)}\}_{n=1}^{N_m}$. Let the index set

$$\mathcal{I}_m = \{n : d_n^{(m)} \neq d_{n'}^{(m)} \ \forall n' < n\} \tag{2.22}$$

The set $\mathcal{I}_m$ contains indices of latent indicators corresponding to unique $Y_{m,i}$, and represent the index values of all components of the stick-breaking construction of G'_m that are used by observations in task m. The known values of $Y_{m,i}$ are therefore $\{c_n^{(m)}\}_{n \in \mathcal{I}_m}$. Let $K := \max_m \max_{n \in \mathcal{I}_m} c_n^{(m)}$. Then the posteriors of $V_1, \ldots, V_K$ are

$$V_j \sim \text{Beta}\left(1 + \sum_{m=1}^{M} \sum_{n \in \mathcal{I}_m} \mathbb{I}(c_n^{(m)} = j), \alpha + \sum_{m=1}^{M} \sum_{n \in \mathcal{I}_m} \mathbb{I}(c_n^{(m)} > j)\right) \tag{2.23}$$

To calculate the posterior of α, values of $V_1, \ldots, V_K$ can be drawn and

$$\alpha \sim \text{Gamma}\left(a_0 + K, b_0 - \sum_{j=1}^{K} \ln(1 - V_j)\right) \tag{2.24}$$

Though the above sampling procedure and that of the previous section may appear complicated, inference using this method is fast, requiring in general less than one second of inference time per iteration.

To test this inference procedure, we ran 5000 experiments using synthesized data. For each trial, we independently sampled the true underlying values $\alpha \sim \text{Uniform}(1, 10)$ and $\beta \sim \text{Uniform}(1, 10)$. We then generated values for $g_0 = (w_1, \ldots, w_K)$ according to (2.20), where K is random and is equal to the smallest value such that $\epsilon < 10^{-6}$. We then generated $M = 10$ group-level probability vectors, $\boldsymbol{\pi}^{(m)} \overset{iid}{\sim} \text{Dir}(\beta g_0)$, and sampled $N_m = 500$ integer-valued samples, $c_n^{(m)} \overset{iid}{\sim} \text{Mult}(\{1, \ldots, K\}, \boldsymbol{\pi}^{(m)})$. For inference, we initialized $\alpha = \beta = 1$ and g_0 to be uniform. We then ran 25 iterations and used the last sample as the inferred value. In actual Gibbs sampling applications, these samples are drawn once, and the "burn-in" phase is concurrent with learning the rest of the model parameters.

In Figure 2.3 we show results for learning α and β using the proposed method. Figure 2.3a shows a scatter plot of the true α versus the learned α. In Figure 2.3b, these values are shown for β. For α, or the top-level scaling parameter, the issue encountered in the previous section does not occur, since the underlying structure of the latent variables is fundamentally different, and there cannot be the large gaps between used components. The underlying structure of the second-level DPs does share this property, however, which accounts for the inflation of the inferred value. The averaging over multiple groups appears to mitigate this problem somewhat, since the inflation is not as large as in the previous section.

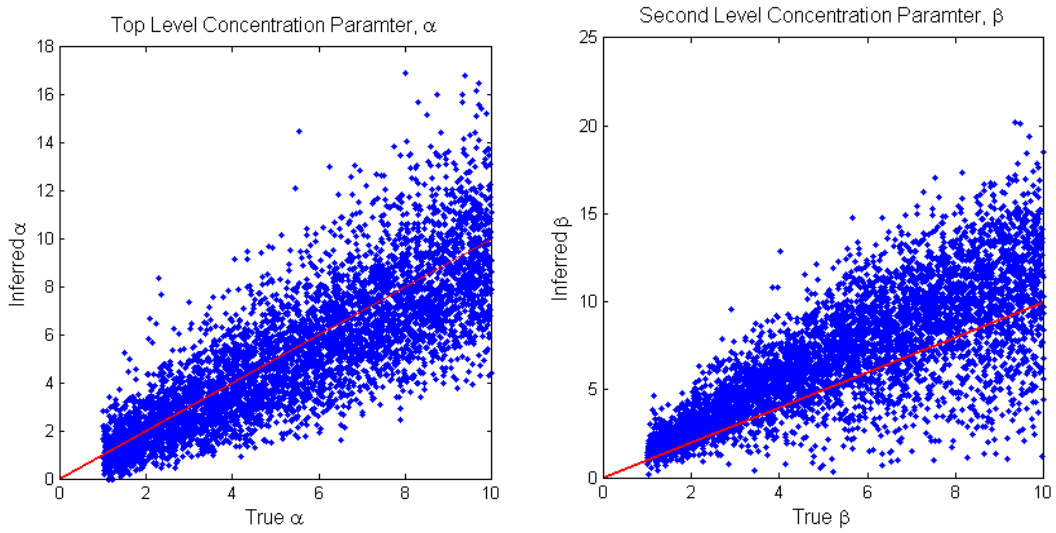


FIGURE 2.3: Learning the concentration parameters of the hierarchical Dirichlet process using Sethuraman's construction. Values for α and β were randomly generated and the probability vector $g_0 = (w_1, \ldots, w_K)$ was generated from a stick-breaking construction using the generated value of α and truncated at $\epsilon < 10^{-6}$. A total of 5000 trials are shown for (left) the inferred values of the top-level concentration parameter, and (right) the second-level concentration parameter.

In Figure 2.4 we show a histogram for the 5000 trials of the L_1 distance between the true underlying g_0 and the inferred g_0. The maximum value of this distance is equal to two, and the histogram of these values shows the inferred distances to be relatively small.

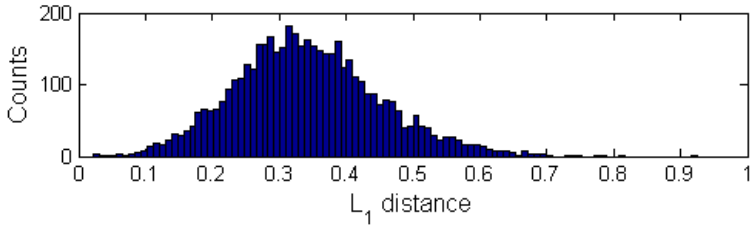


FIGURE 2.4: A histogram of the L_1 distance between the true $g_0 = (w_1, \ldots, w_K)$ and the sampled values of this vector for 5000 trials. The maximum value of this distance is two.

2.4 Appendix

Proof that $\mathbb{E}\left[\epsilon_{\text{DD}}|\alpha, K\right] = \frac{\alpha}{\alpha+K}$: Define the set

$$\mathcal{E}_{\text{DD}} = \{i : Y_i = Y_j \text{ for some } j < i\} \tag{2.25}$$

Then

$$\begin{aligned}
\mathbb{E}\left[\epsilon_{\text{DD}}|\alpha, K\right] &= \mathbb{E}\left[\sum_{i=1}^{\infty} V_i \prod_{j=1}^{i-1}(1-V_j)\mathbb{I}(i \in \mathcal{E}_{\text{DD}})\right] && (2.26)\\
&= \sum_{i=1}^{\infty} \mathbb{E}\left[V_i \prod_{j=1}^{i-1}(1-V_j)\right] \mathbb{P}(i \in \mathcal{E}_{\text{DD}}) && (2.27)\\
&= \sum_{i=1}^{\infty} \frac{\alpha^{i-1}}{(1+\alpha)^i}\left(1 - \left(\frac{K-1}{K}\right)^{i-1}\right) && (2.28)\\
&= 1 - \frac{1}{1+\alpha}\sum_{i=0}^{\infty}\left(\frac{\alpha(K-1)}{(1+\alpha)K}\right)^i && (2.29)\\
&= \frac{\alpha}{\alpha+K} && (2.30)
\end{aligned}$$

The expectation can be brought within the sum by monotone convergence. In (2.28), we use the identity $\mathbb{P}(i \in \mathcal{E}_{\text{DD}}) = 1 - \mathbb{P}(i \notin \mathcal{E}_{\text{DD}})$. We use the geometric series identity to go from (2.29) to (2.30).

3

Dirichlet Processes with Product Base Measures

3.1 Abstract

We have shown how the Dirichlet process can be used as a nonparametric prior for an infinite-dimensional probability mass function on the parameter space of a mixture model. As discussed thus far, the set of parameters over which it is defined is used for a single, parametric distribution. In this chapter, we extend this idea to parameter spaces that characterize multiple distributions, or modalities. In this framework, mixture modeling is performed using observations that contain multiple, incompatible pieces of information, which allows for all information to inform the final clustering result. We provide a general MCMC sampling scheme similar to that in Section 1.6, and demonstrate this framework on a Gaussian-HMM mixture model applied to synthetic and Major League Baseball data. A second application of this framework is also presented in this dissertation, but since it uses the beta process, discussion of this model is deferred to Chapter 6.

3.2 Introduction

The Dirichlet process [27] has proven useful in the machine learning and signal processing communities [80][63][70] as a Bayesian nonparametric prior for mixture models [5]. The infinite support of the Dirichlet process allows for a robust prior definition on a continuous parameter space, and can accommodate an unlimited number of components. Many developments of this framework have been proposed in the literature, e.g. [65][70][80], that vary or add to elements of the generative process. Each addresses a potential aspect of mixture modeling not accounted for in the standard DP, but easily handled via slight modifications. We present here our own modification that accounts for the desire to model data sets where each observation is *itself* a data set of multiple modalities, i.e., multiple statistically irreducible distribution functions, $F^{(m)}(\theta)$. In this case, each observations, X_i, is actually a set of observations, with each part contributing to the characterization of the object of interest. In such cases where multiple pieces of information are available with which objects can be clustered, it is useful to modify the Dirichlet process to account for *all* information when partitioning data into groups. We call this general framework a *Dirichlet process with product base measure* (DP-PBM) as it requires multiple base measures combined in product form to parameterize the Dirichlet process.

This chapter is organized as follows: In Section 2 we present the DP-PBM framework and discuss some of its theoretical properties. In Section 3, we outline a general MCMC inference algorithm for DP-PBM mixture models. Experimental results are given in Section 4, where we focus on a Gaussian-HMM mixture model – one instantiation of the DP-PBM framework. Results are shown for both synthesized and Major League Baseball data for the 2007 season. In Chapter 6, we will show an application of this framework to image interpolation, where the second modality will introduce spatial dependence to the prior.

3.3 The Dirichlet Process with Product Base Measure

In this section, we discuss a variant of the Dirichlet process that incorporates a product base measure, called a DP-PBM, where rather than drawing parameters for one parametric distribution, $\theta \sim G_0$, parameters are drawn for *multiple* distributions, $\theta_m \sim G_{0,m}$ for $m = 1, \ldots, M$. In other words, rather than having a connection between data, $\{X_i\}_{i=1}^N$, and their respective parameters, $\{\theta_{c_i}\}_{i=1}^N$, through a density function, $\{f(\theta_{c_i})\}_{i=1}^N$, *sets* of data, $\{X_{1,i}, \ldots, X_{M,i}\}_{i=1}^N$ have respective *sets* of parameters, $\{\theta_{1,c_i}, \ldots, \theta_{M,c_i}\}_{i=1}^N$, used in inherently different and generally incompatible distribution functions, $\{f^{(1)}(\theta_{1,c_i}), \ldots, f^{(M)}(\theta_{M,c_i})\}_{i=1}^N$.

The DP-PBM is so-called because it utilizes a product base measure to achieve this end, $G_0 = G_{0,1} \times G_{0,2} \times \cdots \times G_{0,M}$, where in this case, M modalities are considered. The space over which this process is defined is now $\left(\prod_{m=1}^M \Theta_m, \bigotimes_{m=1}^M \mathcal{B}_m, \prod_{m=1}^M G_{0,m}\right)$. Though this construction implicitly takes place in all mixture models that attempt to estimate multiple parameters, for example the multivariate Gaussian mixture model, we believe this general framework of using these parameters in multiple, incompatible distributions (or modalities) is novel. The full generative process can be written as follows:

$$
\begin{aligned}
X_{m,i} &\sim f^{(m)}(\theta_{m,c_i}) && (3.1)\\
c_i &\overset{iid}{\sim} \text{Mult}(\{1, \ldots, K\}, \boldsymbol{\pi}) && (3.2)\\
\pi_j &= V_j \prod_{\ell=1}^{j-1} (1 - V_\ell) && (3.3)\\
V_j &\overset{iid}{\sim} \text{Beta}(1, \alpha) && (3.4)\\
\theta_{m,j} &\sim G_{0,m} && (3.5)
\end{aligned}
$$

for $m = 1, \ldots, M$, where $\theta_{m,j}$ are drawn iid from $G_{0,m}$ for a fixed m and independently under varying m. Note that if each $G_{0,m}$ is a univariate normal-gamma prior, this

model reduces to a multivariate GMM with a forced diagonal covariance matrix. As previously stated, we are more interested in cases where each X_m is inherently incompatible, but is still linked by the structure of the data set.

For example, consider a set of observations, $\{O_i\}_{i=1}^N$, where each $O_i = \{X_{1,i}, X_{2,i}\}$ with $X_1 \in \mathbb{R}^d$ and X_2 a sequence of time-series data. In this case, a single density function, $f(X_1, X_2|\theta_1, \theta_2)$ cannot analytically accommodate O_i, making inference difficult. However, if these densities can be considered as *independent*, that is $f(X_1, X_2|\theta_1, \theta_2) = f(X_1|\theta_1) \cdot f(X_2|\theta_2)$, then this problem becomes analytically tractable and, furthermore, no more difficult to solve than for the standard Dirichlet process. One might choose to model X_1 with a Gaussian distribution, with $G_{0,1}$ the appropriate prior and X_2 by an HMM [61], with $G_{0,2}$ its respective prior. In this case, this model becomes a hybrid Gaussian-HMM mixture, where each component is *both* a Gaussian *and* a hidden Markov model.

3.3.1 Predicting Values for Missing Modalities

As alluded to in the previous section, the analytical nature of the DP-PBM framework depends upon a factorization of the likelihood function. That is, for the likelihood function of our M-modality data, we assume that we can write

$$f(X_1, \ldots, X_M|\theta_1, \ldots, \theta_M) = \prod_{m=1}^{M} f^{(m)}(X_m|\theta_m)$$

where $f^{(m)}(X_m|\theta_m)$ is the likelihood function and θ_m the parameter (or set of parameters) for the m^{th} modality. As will be seen in the next section, inference then becomes analytical, provided the appropriate priors, $p(\theta_m)$, are selected. This is because the difference modalities are all independent conditioned upon the latent indicator, c, which selects the set of parameters for all M distribution functions.

Because of this independence assumption, it might seem that the model will not capture any correlations within the data across modalities. While it is true that this

ability is not given to the prior, the *posterior* will capture correlations. For example, given the posterior for N observations, consider an $N+1^{st}$ observation where the first $M-1$ modalities are present, but the M^{th} is missing. If we wish to infer its associated latent indicator, c_{N+1} (or which component it came from), we can simply calculate for the first $M-1$ modalities

$$P(c_{N+1} = j|\boldsymbol{X}, \boldsymbol{\theta}) \propto \pi_j \prod_{m=1}^{M-1} f^{(m)}(x_{m,N+1}|\theta_{m,j}) \tag{3.6}$$

effectively integrating out the M^{th} modality. Here, θ_{m_j} can be a sample from its posterior, or this parameter can be integrated out, in which case the conditioning is on the posterior parameters of θ_{m_j}. If integration is intractable, Monte Carlo integration methods can also be used [29]. We see that, given the distribution on c_{N+1}, we can then interpolate or make predictions as to the missing modality, $x_{M,N+1}$, by forming a mixture using the probabilities in (3.6) for the mixing weights and samples from the respective posteriors of the parameters of the missing modality. These parameters can also be integrated out, if tractable, allowing for sampling of $x_{M,N+1}$ from the posterior marginal distribution.

3.4 MCMC Inference for DP-PBM Mixture Models

In this section, we outline a general method for performing Markov chain Monte Carlo (MCMC) [29] inference for DP-PBM models. We let $f^{(m)}(x_m|\theta_m)$ be the likelihood function for the m^{th} modality of an observation given the parameters, θ_m, and let $p(\theta_m)$ the prior density of θ_m. We refer to the DP-PBM as G, where $G = \sum_{j=1}^{K+1} \pi_j \delta_{\{\theta_{m,j}\}_{m=1}^{M}}$. We also observe that this sampling method is unbounded in the potential number of components, but only requires the K occupied components plus a $K+1^{st}$ proposal component for any given iteration, as in Section 1.6.

Initialization: Select a truncation level, $K+1$, and initialize the model, G, by

sampling $\theta_{m,k} \sim G_{m,0}$ for $k = 1, \ldots, K+1$, $m = 1, \ldots, M$ and $V_k \sim Beta(1, \alpha)$ for $k = 1, \ldots, K$ and construct $\boldsymbol{\pi} = \phi_K(\boldsymbol{V})$.

Step 1: Sample the indicators, $c_1, \ldots, c_N$, independently from their respective conditional posteriors, $p(c_j | \{x_{m,1}\}_{m=1}^M) \propto \prod_{m=1}^M f^{(m)}(x_{m,j}|\theta_{m,c_j}) p(\theta_{m,c_j}|G)$,

$$c_j \sim \sum_{k=1}^{K+1} \frac{\pi_k \prod_{m=1}^M f^{(m)}(x_{m,j}|\theta_{m,k})}{\sum_\ell \pi_\ell \prod_{m=1}^M f^{(m)}(x_{m,j}|\theta_{m,\ell})} \delta_k \tag{3.7}$$

Set K to be the number of unique values among $c_1, \ldots, c_N$ and relabel from 1 to K.

Step 2: Sample $\{\theta_{m,1}\}_{m=1}^M, \ldots, \{\theta_{m,K}\}_{m=1}^M$ from their respective posteriors conditioned on $c_1, \ldots, c_N$ and $x_1, \ldots, x_N$,

$$\theta_{m,k} \sim p\left(\theta_{m,k} | \{c_j\}_{j=1}^N, x_{m,1}, \ldots, x_{m,N}\right) \tag{3.8}$$

$$p\left(\theta_{m,k} | \{c_j\}_{j=1}^N, x_{m,1}, \ldots, x_{m,N}\right) \propto \prod_{j=1}^N f^{(m)}(x_{m,j}|\theta_m)^{\mathbb{I}(c_j=k)} p(\theta_m) \tag{3.9}$$

where $\mathbb{I}(c_j = k)$ equals one if $c_j = k$ and zero otherwise; this simply picks out which $\{x_{m,j}\}_{m=1}^M$ belong to component k. Sample $\theta_{m,K+1} \sim G_{0,m}$ for $m = 1, \ldots M$. These M posteriors are calculated independently of one another given the relevant data for that modality extracted from the observations assigned to that component. We stress that when an "observation" is assigned to a component (via the indicator, c) it is actually *all* of the data that comprise that observation that is being assigned to the component.

Step 3: Construct the $(K+1)$-dimensional weight vector, $\boldsymbol{\pi} = \phi_K(\boldsymbol{V})$, using $V_1, \ldots, V_K$ sampled from their beta-distributed posteriors conditioned on $c_1, \ldots, c_N$,

$$V_k \sim \text{Beta}\left(1 + \sum_{j=1}^N \mathbb{I}(c_j = k), \alpha + \sum_{\ell=k+1}^K \sum_{j=1}^N \mathbb{I}(c_j = \ell)\right) \tag{3.10}$$

Set $\pi_{K+1} = \prod_{k=1}^{K}(1 - V_k)$.

Repeat Steps 1 – 3 for a desired number of iterations. The convergence of this Markov chain can be assessed [29], after which point uncorrelated samples (properly spaced out in the chain) of the values in Steps 1 – 3 are iid samples from the full posterior of the model parameters. As can be seen, inference for DP-PBM models is straightforward and, when each $p(\theta_m)$ is conjugate to $f^{(m)}(x_m|\theta_m)$, fully analytical.

3.5 Applications: The Gaussian-HMM Mixture Model

We look at a concrete example of a DP-PBM model, a Gaussian-HMM mixture model, where modality one is data $X_1 \in \mathbb{R}^d$ and modality two is a sequence drawn from a hidden Markov model [61], $X_2 \sim \text{HMM}(\mathbf{A}, \mathbf{B}, \pi')$. Our experiments are performed on synthesized and Major League Baseball (MLB) data sets.

3.5.1 Experiment with Synthesized Data

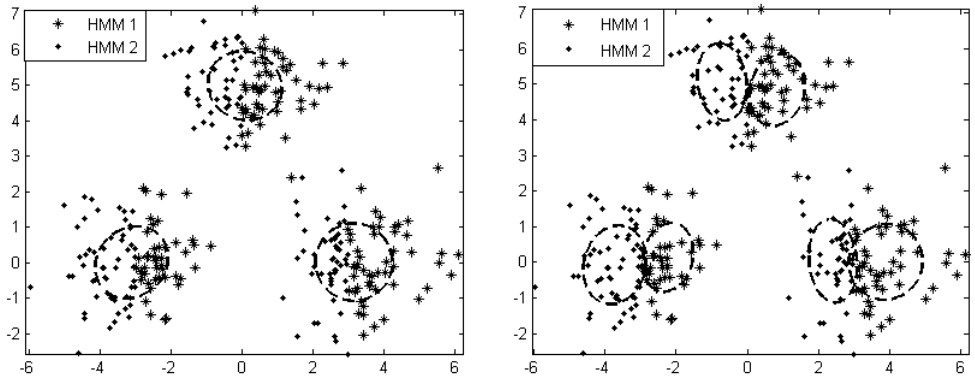
We define three, two-dimensional Gaussian distributions with respective means $\mu_1 = (-3, 0)$, $\mu_2 = (3, 0)$ and $\mu_3 = (0, 5)$ and each having the identity as the covariance matrix. Two hidden Markov models are defined as below,

$$\mathbf{A_1} = \begin{bmatrix} 0.05 & 0.9 & 0.05 \\ 0.05 & 0.05 & 0.9 \\ 0.9 & 0.05 & 0.05 \end{bmatrix} \quad \mathbf{A_2} = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.05 & 0.9 & 0.05 \\ 0.05 & 0.05 & 0.9 \end{bmatrix}$$

$$\mathbf{B_1}, \mathbf{B_2} = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.05 & 0.9 & 0.05 \\ 0.05 & 0.05 & 0.9 \end{bmatrix}$$

with the initial state vector $\boldsymbol{\pi'}_1, \boldsymbol{\pi'}_2 = [1/3, 1/3, 1/3]$. Data was generated as follows: We sampled 300 observations, 100 from each Gaussian, constituting $X_{1,i}$ for $i = 1, \ldots, 300$. For each sample, if the observation was on the right half of its respective Gaussian, a sequence of length 50 was drawn from HMM 1, if on the left, from HMM 2. For display purposes, we select a typical sample from MCMC inference.

This precisely defined data set allows the model to clearly display the benefits of its design. If one were to build a Gaussian mixture model on the X_1 data alone, three components would be uncovered, as shown in Figure 1(a). If an HMM mixture were built alone on the X_2 data, only two components would be uncovered. Using *all* of the data, that is, mixing on $\{O_i\}_{i=1}^{300}$ rather than just $\{X_{1,i}\}_{i=1}^{300}$ or $\{X_{2,i}\}_{i=1}^{300}$ alone, the correct number of six components was uncovered, as shown in Figure 1(b).



(*a*) (*b*)

FIGURE 3.1: An example of a mixed Gaussian-HMM data set. (a) Gaussian mixture model results. (b) Gaussian-HMM mixture results. Each ellipse corresponds to a cluster.

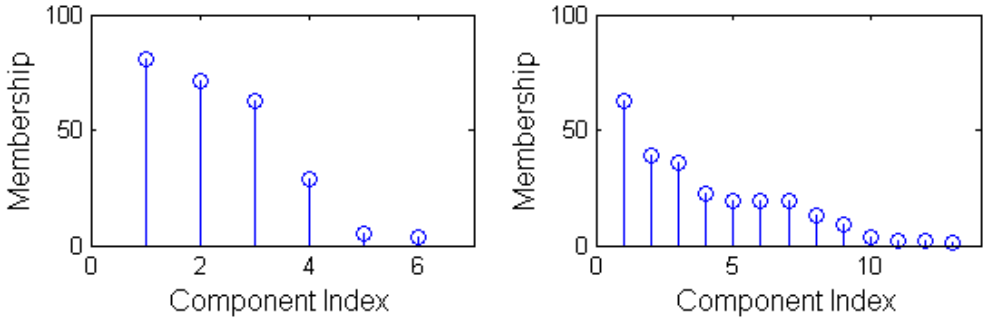
The results show that, as was required by the data, the DP-PBM prior uncovered six distinct clusters of data. The DP-PBM framework allowed for the incorporation of *all* information of the data set to be included, thus providing more precise clustering results.

3.5.2 Major League Baseball Data Set

Using the complete bat-by-bat statistics for the 2007 season[1], we processed our data set as follows. We created a 3-dimensional vector, X_1, of the batting average, on-base percentage and slugging percentage. We then quantized the plate appearances for a given player into the following codes: 1. Strikeout, 2. Fielded out, 3. Hit, where

[1] Data was obtained from www.retrosheet.org

walks and other results were ignored. We limited our set to the 252 players with a sequence length greater than 300. For MCMC, we used 1000 burn-in and 3000 burn-out iterations and selected an iteration of median likelihood for presentation below. We also show results for an HMM mixture model [60] without using the spatial data. The component membership results, or the number of observations that were assigned to a given indexed component, are shown in Figure 2 for both models. We see that using additional information produces a more refined clustering result, as was the case in the synthetic result.



(a) (b)

FIGURE 3.2: Component membership results for MLB data when (a) X_1 data is ignored – the HMM mixture model. (b) both X_1 and X_2 data is used – the Gaussian-HMM mixture model.

We next ask whether the increase in the number of clusters results in a more precise and informative clustering result. To do this we consider two measures, first the average differential entropy of the Gaussian component, where we empirically calculated the covariance from the HMM mixture results.

$$h_{avg}(X_1) = \sum_{i=1}^{K} \pi_i \frac{1}{2} \ln\left((2\pi e)^3 |\Sigma_i|\right) \tag{3.11}$$

We recall that differential entropy can be negative and that $h_{avg}(X_1) \to -\infty$ as the uncertainty tends to zero. Using this measure for the HMM mixture, $h_{avg}(X_1) = -6.31$, while for the Gaussian-HMM mixture, $h_{avg}(X_1) = -7.11$, indicating that the

Gaussian-HMM more precisely represented the spatial information, thus improving clustering.

As a second measure, we consider the average entropy of each HMM, which is estimated using the original data

$$H_{avg}(X_2) = -\sum_{i=1}^{K} \pi_i \sum_{n=1}^{N_i} \frac{1}{N_i} \ln P(X_{2,\rho_i(n)}|A_i, B_i, \pi_i') \tag{3.12}$$

where N_i is the number of data in component i, with $\rho_i(n)$ selecting the appropriate X_2. Using this measure, for the HMM mixture we found that, $H_{avg}(X_2) = 477.4$, and for the Gaussian-HMM mixture, $H_{avg}(X_2) = 476.7$. Therefore, performance for the HMM is comparable. This is reasonable when viewed in light of the synthetic example. We've therefore seen that clustering with all data tends to improve the overall result as it refines the clustering in a meaningful way.

3.6 Conclusions

In this chapter, we have derived an extension of the Dirichlet process that mixes on all data in an observation by using a product base distribution that allows for multiple modalities in a data set. As an example, we developed the Gaussian-HMM mixture model, where each component generated data from both a multivariate Gaussian distribution and a hidden Markov model. Experimental results showed the functioning of this model on both synthesized and MLB data for clustering. In Chapter 6, we will show an application of this framework to image interpolation, which uses a beta process factor analysis model as one modality. The second modality allows for spatial information to be exploited within the image.

4

The Beta Process for Latent Factor Models

4.1 Abstract

In this chapter, we propose a nonparametric extension to the factor analysis problem using a beta process prior. This *beta process factor analysis* (BPFA) model allows for a dataset to be decomposed into a linear combination of a sparse set of factors, providing information on the underlying structure of the observations. As with the Dirichlet process, the beta process is a fully Bayesian conjugate prior, which allows for analytical posterior calculation and straightforward inference. We derive a variational Bayes inference algorithm and demonstrate the model on the MNIST digits and HGDP-CEPH cell line panel datasets. We also present results for an application to basis design for compressive sensing.

4.2 Introduction

Latent membership models provide a useful means for discovering underlying structure in a dataset by elucidating the relationships between observed data. For example, in *latent class models*, observations are assumed to be generated from one

of K classes, with mixture models constituting a classic example. When a single class indicator is considered too restrictive, *latent feature models* can be employed, allowing for an observation to possess combinations of up to K latent features.

As K is typically unknown, Bayesian nonparametric models seek to remove the need to set this value by defining robust, but sparse priors on infinite spaces. For example, the Dirichlet process [27] allows for nonparametric mixture modeling in the latent class scenario. In the latent feature paradigm, the beta process [34] has been defined and can be used toward the same objective, which, when marginalized, is closely related to the Indian buffet process [31, 73].

An example of a latent feature model is the factor analysis model [76], where a data matrix is decomposed into the product of two matrices plus noise,

$$X = \Phi Z + E \tag{4.1}$$

In this model, the columns of the $D \times K$ matrix of factor loadings, Φ, can be modeled as latent features and the elements in each of N columns of Z can be modeled as indicators of the possession of a feature for the corresponding column of X (which can be given an associated weight). It therefore seems natural to seek a nonparametric model for this problem.

To this end, several models have been proposed that use the Indian buffet process (IBP) [44, 62, 47]. However, these models require MCMC inference, which can be slow to converge. In this chapter, we propose a *beta process factor analysis* (BPFA) model that is fully conjugate and therefore has a fast variational solution; this is an intended contribution of this chapter. Starting from first principles, we show how the beta process can be formulated to solve the nonparametric factor analysis problem, as the Dirichlet process has been previously shown to solve the nonparametric mixture modeling problem; we intend for this to be a second contribution of this chapter.

The remainder of this chapter is organized as follows. In Section 4.3 we review the

beta process in detail. We introduce the BPFA model in Section 4.4, and discuss some of its theoretical properties. In Section 4.5 we derive a variational Bayes inference algorithm for fast inference, exploiting full conjugacy within the model. Experimental results are presented in Section 4.6 on synthetic data, and on the MNIST digits and HGDP-CEPH cell line panel [67] datasets. We also present results for dictionary learning, to be used in compressive sensing applications. We conclude and discuss future work in Section 4.7.

4.3 The Beta Process

The beta process, first introduced by Hjort for survival analysis [34], is an independent increments, or Lévy process and can be defined as follows:

Definition: Let Ω be a measurable space and $\mathcal{B}$ its σ-algebra. Let H_0 be a continuous measure on $(\Omega, \mathcal{B})$ with $H_0(\Omega) = \gamma$, and let α be a positive scalar. Then for all disjoint, infinitesimal partitions, $\{B_1, \ldots, B_K\}$, of Ω the beta process is generated as follows,

$$H(B_k) \sim \text{Beta}(\alpha H_0(B_k), \alpha(1 - H_0(B_k))) \tag{4.2}$$

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

Because of the convolution properties of beta random variables, the beta process does not satisfy the Kolmogorov consistency condition, and is therefore defined in the infinite limit [34]. Hjort extends this definition to include functions, $\alpha(B_k)$, which for simplicity is here set to a constant.

Like the Dirichlet process, the beta process can be written in set function form,

$$H(\omega) = \sum_{k=1}^{\infty} \pi_k \delta_{\omega_k}(\omega) \tag{4.3}$$

with $H(\omega_i) = \pi_i$. Also like the Dirichlet process, means for drawing H are not

obvious. We briefly discuss this issue in Section 4.3.1. In the case of the beta process, π does not serve as a probability mass function on Ω, but rather as part of a new measure on Ω that parameterizes a Bernoulli process defined as follows:

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

$$z_{ik} \sim \text{Bernoulli}(\pi_k) \tag{4.4}$$

The new measure, $X_i(\omega) = \sum_k z_{ik}\delta_{\omega_k}(\omega)$, is then drawn from a Bernoulli process, or $X_i \sim \text{BeP}(H)$.

By arranging samples of the infinite-dimensional vector, z_i, in matrix form, $Z = [z_1, \ldots, z_N]$, the beta process is seen to be a prior over infinite binary matrices, with each row in the matrix Z corresponding to a location, δ_ω.

4.3.1 The Marginalized Beta Process and the Indian Buffet Process

As previously mentioned, sampling H directly from the infinite beta process is difficult, but a marginalized approach can be derived in the same manner as the corresponding Chinese restaurant process [3], used for sampling from the Dirichlet process. We briefly review this marginalization, discussing the link to the Indian buffet process [31] as well as other theoretical properties of the beta process that arise as a result.

We first extend the beta process to take two scalar parameters, a, b, and partition Ω into K regions of equal measure, or $H_0(B_k) = 1/K$ for $k = 1, \ldots, K$. We can then write the generative process in the form of (4.3) as

$$\begin{aligned} H(B) &= \sum_{k=1}^{K} \pi_k \delta_{B_k}(B) \\ \pi_k &\sim \text{Beta}\left(a/K, b(K-1)/K\right) \end{aligned} \tag{4.5}$$

where $B \in \{B_1, \ldots, B_K\}$. Marginalizing the vector π and letting $K \to \infty$, the

matrix, Z, can be generated directly from the beta process prior as follows:

1. For an infinite matrix, Z, initialized to all zeros, set the first $c_1 \sim \text{Poisson}(a/b)$ rows of z_1 to 1. Sample the associated locations, ω_i, $i = 1, \ldots, c_1$, independently from H_0.

2. For observation N, sample $c_N \sim \text{Poisson}\left(\frac{a}{b+N-1}\right)$ and define $C_N := \sum_{i=1}^{N} c_i$. For rows $k = 1, \ldots, C_{N-1}$ of z_N, sample

$$z_{Nk} \sim \text{Bernoulli}\left(\frac{n_{Nk}}{b+N-1}\right) \tag{4.6}$$

   where $n_{Nk} := \sum_{i=1}^{N-1} z_{ik}$, the number of previous observations with a 1 at location k. Set indices $C_{N-1}+1$ to C_N equal to 1 and sample associated locations independently from H_0.

If we define

$$H(\omega) := \sum_{k=1}^{\infty} \frac{n_{Nk}}{b+N-1} \delta_{\omega_k}(\omega) \tag{4.7}$$

then $H \sim \text{BP}(a, b, H_0)$ in the limit as $N \to \infty$, and the exchangeable columns of Z are drawn iid from a beta process. As can be seen, in the case where $b = 1$, the marginalized beta process is equivalent to the Indian buffet process [73].

This representation can be used to derive some interesting properties of the beta process. We observe that the random variable, C_N, has a Poisson distribution, $C_N \sim \text{Poisson}\left(\sum_{i=1}^{N} \frac{a}{b+i-1}\right)$, which provides a sense of how the matrix Z grows with sample size. Furthermore, since $\sum_{i=1}^{N} \frac{a}{b+i-1} \to \infty$ as $N \to \infty$, we can deduce that the entire space of Ω will be explored as the number of samples grows to infinity.

We show in Figure 4.1 the expectation of π calculated empirically by drawing from the marginalized beta process. As can be seen, the a, b parameters offer flexibility in both the magnitude and shape of π and can be tuned.

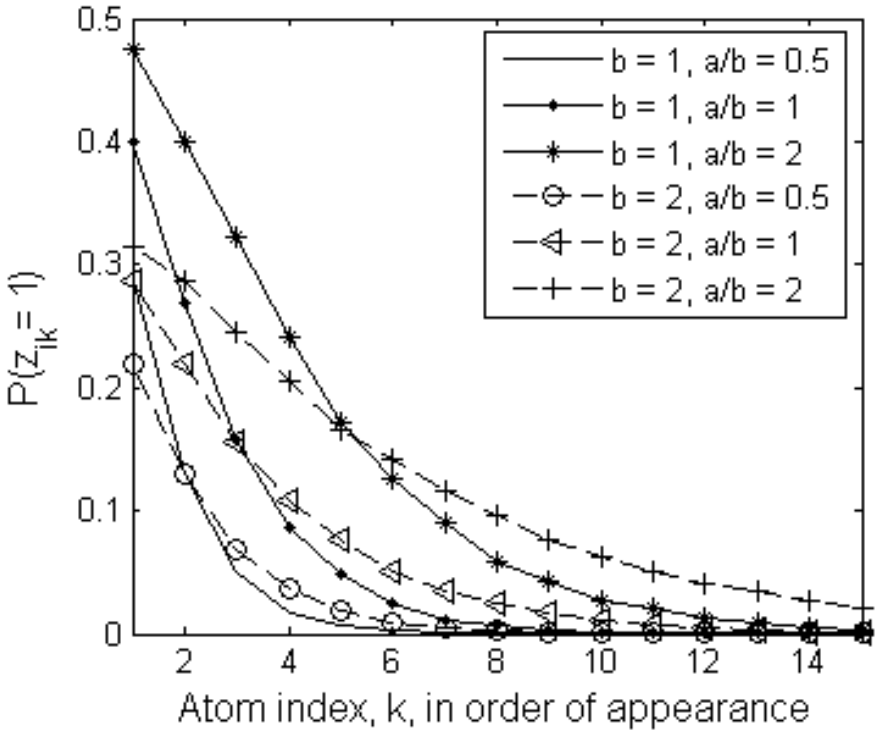


FIGURE 4.1: Estimation of π from 5000 marginal beta process runs of 500 samples each, with various a, b initializations.

4.3.2 Finite Approximation to the Beta Process

As hinted in (4.5), a finite approximation to the beta process can be made by simply setting K to a large, but finite number. This approximation can be viewed as serving a function similar to the finite Dirichlet distribution in its approximation of the infinite Dirichlet process for mixture modeling. The finite representation is written as

$$\begin{aligned} H(\omega) &= \sum_{k=1}^{K} \pi_k \delta_{\omega_k}(\omega) \\ \pi_k &\sim \text{Beta}(a/K, b(K-1)/K) \\ \omega_k &\overset{iid}{\sim} H_0 \end{aligned} \tag{4.8}$$

with the K-dimensional vector, z_i, drawn from a finite Bernoulli process parameterized by H. The full conjugacy of this representation means posterior computation is analytical, which will allow for variational inference to be performed on the BPFA model.

We briefly mention that a stick-breaking construction of the beta process has recently been derived by the author and will be presented in the next chapter; this allows for exact Bayesian inference. A construction for the Indian buffet process has also been presented [71], though this method does not extend to the more general beta process. We will use the finite approximation presented here in the following sections.

4.4 Beta Process Factor Analysis

Factor analysis can be viewed as the process of modeling a data matrix, $X \in \mathbb{R}^{D\times N}$, as the multiplication of two matrices, $\Phi \in \mathbb{R}^{D\times K}$ and $Z \in \mathbb{R}^{K\times N}$, plus an error matrix, E.

$$X = \Phi Z + E \tag{4.9}$$

Often, prior knowledge about the structure of the data is used, for example, the desired sparseness properties of the Φ or Z matrices [76, 62, 44]. The beta process is another such prior that achieves this sparseness, allowing for K to tend to infinity while only focusing on a small subset of the columns of Φ via the sparse matrix Z.

In *beta process factor analysis* (BPFA), we model the matrices Φ and Z as N draws from a Bernoulli process parameterized by a beta process, H. First, we recall that draws from the BeP-BP approximation can be generated as

$$\begin{aligned} z_{ik} &\sim \text{Bernoulli}(\pi_k) \\ \pi_k &\sim \text{Beta}(a/K, b(K-1)/K) \\ \phi_k &\overset{iid}{\sim} H_0 \end{aligned} \tag{4.10}$$

for observation $i = 1, \ldots, N$ and latent feature (or factor) $k = 1, \ldots, K$. In the general definition, H_0 was unspecified, as was the use of the latent membership vector, z_i. For BPFA, we let H_0 be multivariate normal and the latent factors be indicators of linear combinations of these locations, which can be written in matrix notation as Φz_i, where $\Phi = [\phi_1, \ldots, \phi_K]$. Adding the noise vector, ϵ_i, we obtain observation x_i. The beta process can thus be seen as a prior on the parameters, $\{\pi, \Phi\}$, with iid Bernoulli process samples composing the expectation matrix, $\mathbb{E}[X] = \Phi Z$ for the factor analysis problem.

As an unweighted linear combination might be too restrictive, we include a weight vector, w_i, which results in the following generative process for observation $i = 1, \ldots, N$,

$$\begin{aligned} x_i &= \Phi(z_i \circ w_i) + \epsilon_i \\ w_i &\sim \mathcal{N}(0, \sigma_w^2 I) \\ z_{ik} &\sim \text{Bernoulli}(\pi_k) \\ \pi_k &\sim \text{Beta}(a/K, b(K-1)/K) \\ \phi_k &\sim \mathcal{N}(0, \Sigma) \\ \epsilon_i &\sim \mathcal{N}(0, \sigma_n^2 I) \end{aligned} \tag{4.11}$$

for $k = 1, \ldots, K$ and all values drawn independently. The symbol $\circ$ represents the Hadamard, or elementwise multiplication of two vectors. We show a graphical representation of the BPFA model in Figure 4.2.

Written in matrix notation, the weighted BP-F model of (4.11) is thus a prior on

$$X = \Phi(Z \circ W) + E \tag{4.12}$$

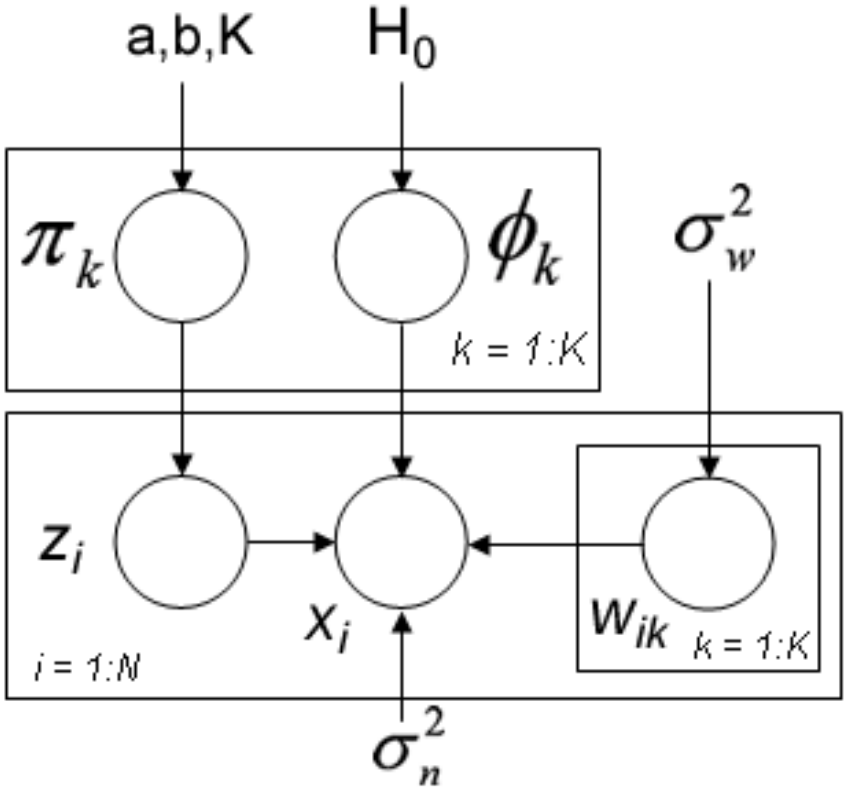


FIGURE 4.2: A graphical representation of the BPFA model.

Under this prior, the mean and covariance of a given vector, x, can be calculated,

$$\begin{aligned} \mathbb{E}[x] &= 0 \\ \mathbb{E}[xx^T] &= \frac{aK}{a+b(K-1)}\sigma_w^2\Sigma + \sigma_n^2 I \end{aligned} \tag{4.13}$$

Letting $K \to \infty$, we see that $\mathbb{E}[xx^T] \to \frac{a}{b}\sigma_w^2\Sigma + \sigma_n^2 I$. Therefore, the BPFA model remains well-defined in the infinite limit. To emphasize, compare this value with z removed, where $\mathbb{E}[xx^T] = K\sigma_w^2\Sigma + \sigma_n^2 I$. The coefficient $\frac{a}{b}$ is significant in that it represents the expected number of factors present in an observation as $K \to \infty$. That is, if we define $m_i := \sum_{k=1}^{\infty} z_{ik}$, where $z_i \sim \text{BeP}(H)$ and $H \sim \text{BP}(a, b, H_0)$, then by marginalizing H we find that $\mathbb{E}[m_i] = \frac{a}{b}$.

Another important aspect of the BPFA model is that the π vector enforces sparse-

ness on the *same* subset of factors. In comparison, consider the model where z_i is removed and sparseness is enforced by sampling the elements of w_i iid from a sparseness inducing normal-gamma prior. This is equivalent to learning multiple relevance vector machines [75] with a jointly learned and shared Φ matrix. A theoretical issue with this model is that the prior does not induce sparseness on the *same* subset of latent factors. As $K \to \infty$, all factors will be used sparsely with equal probability and, therefore, no factors will be shared. This is conceptually similar to the problem of drawing multiple times from a Dirichlet process prior, where individual draws are sparse, but no two draws are sparse on the same subset of atoms. We note that the hierarchical Dirichlet process has been introduced to resolve this particular issue [70].

4.5 Variational Bayesian Inference

In this section, we derive a variational Bayesian algorithm [8, 77] to perform fast inference for the weighted BPFA model of (4.11). This is aided by the conjugacy of the beta to the Bernoulli process, where the posterior for the single parameter beta process is

$$H|X_1,\ldots,X_N \sim BP\left(\alpha H_0 + \sum_{i=1}^{N} X_i\right) \tag{4.14}$$

with $X_i \sim \text{BeP}(H)$ being the i^{th} sample from a Bernoulli process parameterized by H. The two-parameter extension has a similar posterior update, though not as compact a written form.

In the following, we define $x_i^{-k} := x_i - \Phi_{-k}(z_i^{-k} \circ w_i^{-k})$, where Φ_{-k}, z^{-k} and w^{-k} are the matrix/vectors with the k^{th} column/element removed; this is simply the portion of x_i remaining considering all but the k^{th} factor. Also, for clarity, we have suppressed certain equation numbers and conditional variables.

4.5.1 The VB-E Step

Update for Z: $p(z_{ik}|x_i, \Phi, w_i, z_i^{-k}) \propto p(x_i|z_{ik}, \Phi, w_i, z_i^{-k})p(z_{ik}|\pi)$

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

$\exp[\langle \ln(\pi_k) \rangle] \times$

$$\exp\left[-\frac{1}{2\sigma_n^2}\left(\langle w_{ik}^2 \rangle \langle \phi_k^T \phi_k \rangle - 2\langle w_{ik} \rangle \langle \phi_k \rangle^T \langle x_i^{-k} \rangle\right)\right]$$

where $\langle \cdot \rangle$ indicates the expectation. The probability that $z_{ik} = 0$ is proportional to $\exp[\langle \ln(1 - \pi_k) \rangle]$. The expectations can be calculated as

$$\langle \ln(\pi_k) \rangle \;=\; \psi\left(\frac{a}{K} + \langle n_k \rangle\right) - \psi\left(\frac{a + b(K-1)}{K} + N\right)$$

$\langle \ln(1 - \pi_k) \rangle \;=$

$$\psi\left(\frac{b(K-1)}{K} + N - \langle n_k \rangle\right) - \psi\left(\frac{a + b(K-1)}{K} + N\right)$$

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

$$\langle w_{ik}^2 \rangle \;=\; \langle w_{ik} \rangle^2 + \Delta_i'^{(k)} \tag{4.15}$$

$$\langle \phi_k^T \phi_k \rangle \;=\; \langle \phi_k \rangle^T \langle \phi_k \rangle + \mathrm{trace}(\Sigma_k') \tag{4.16}$$

where $\langle n_k \rangle$ is defined in the update for π, Σ_k' in the update for Φ, and $\Delta_i'^{(k)}$ is the k^{th} diagonal element of Δ_i' defined in the update for W.

4.5.2 The VB-M Step

Update for π: $p(\pi_k|Z) \propto p(Z|\pi_k)p(\pi_k|a, b, K)$

The posterior of π_k can be shown to be

$$\pi_k \sim Beta\left(\frac{a}{K} + \langle n_k \rangle, \frac{b(K-1)}{K} + N - \langle n_k \rangle\right)$$

where $\langle n_k \rangle = \sum_{i=1}^{N} \langle z_{ik} \rangle$ can be calculated from the VB-E step. The priors a, b can be tuned according to the discussion in Section 4.3.1. We recall that $\sum_{i=1}^{N} \frac{a}{b+i-1}$ is the expected total number of factors, while a/b is the expected number of factors used by a single observation in the limiting case.

Update for Φ: $p(\phi_k | X, \Phi_{-k}, Z, W) \propto p(X | \phi_k, \Phi_{-k}, Z, W) p(\phi_k | \Sigma)$

The posterior of ϕ_k can be shown to be normal with mean, μ'_k, and covariance, Σ'_k, equal to

$$\Sigma'_k = \left(\frac{1}{\sigma_n^2} \sum_{i=1}^{N} \langle z_{ik} \rangle \langle w_{ik}^2 \rangle I + \Sigma^{-1} \right)^{-1} \tag{4.17}$$

$$\mu'_k = \Sigma'_k \left(\frac{1}{\sigma_n^2} \sum_{i=1}^{N} \langle z_{ik} \rangle \langle w_{ik} \rangle \langle x_i^{-k} \rangle \right) \tag{4.18}$$

with $\langle w_{ik}^2 \rangle$ given in (4.15). The prior Σ can be set to the empirical covariance of the data, X.

Update for W: $p(w_i | x_i, \Phi, z_i) \propto p(x_i | w_i, \Phi, z_i) p(w_i | \sigma_w^2)$

The posterior of w_i can be shown to be multivariate normal with mean, υ'_i, and covariance, Δ'_i, equal to

$$\Delta'_i = \left(\frac{1}{\sigma_n^2} \langle \tilde{\Phi}_i^T \tilde{\Phi}_i \rangle + \frac{1}{\sigma_w^2} I \right)^{-1} \tag{4.19}$$

$$\upsilon'_i = \Delta'_i \left(\frac{1}{\sigma_n^2} \langle \tilde{\Phi}_i \rangle^T x_i \right) \tag{4.20}$$

where we define $\tilde{\Phi}_i := \Phi \circ \tilde{Z}_i$ and $\tilde{Z}_i := [z_i, \ldots, z_i]^T$, with the K-dimensional vector, z_i, repeated D times. Given that $\langle \tilde{\Phi}_i \rangle = \langle \Phi \rangle \circ \langle \tilde{Z}_i \rangle$, we can then calculate

$$\langle \tilde{\Phi}_i^T \tilde{\Phi}_i \rangle = \left(\langle \Phi \rangle^T \langle \Phi \rangle + A \right) \circ \left(\langle z_i \rangle \langle z_i \rangle^T + B_i \right) \tag{4.21}$$

where A and B_i are calculated as follows

$$\begin{aligned} A &:= \operatorname{diag}\left[\operatorname{trace}(\Sigma'_1), \ldots, \operatorname{trace}(\Sigma'_K)\right] \\ B_i &:= \operatorname{diag}\left[\langle z_{i1}\rangle(1-\langle z_{i1}\rangle), \ldots, \langle z_{iK}\rangle(1-\langle z_{iK}\rangle)\right] \end{aligned}$$

A prior, discussed below, can be placed on σ_w^2, removing the need to set this value.

Update for σ_n^2: $p(\sigma_n^2|X,\Phi,Z,W) \propto p(X|\Phi,Z,W,\sigma_n^2)p(\sigma_n^2)$

We can also infer the noise parameter, σ_n^2, by using an inverse-gamma, $InvGa(c,d)$, prior. The posterior can be shown to be inverse-gamma with

$$\begin{aligned} c' &= c + \frac{ND}{2} \qquad (4.22) \\ d' &= d + \frac{1}{2}\sum_{i=1}^{N}\left(\|x_i - \langle\Phi\rangle(\langle z_i\rangle \circ \langle w_i\rangle)\|^2 + \xi_i\right) \end{aligned}$$

where

$$\xi_i := \sum_{k=1}^{K}\left(\langle z_{ik}\rangle\langle w_{ik}^2\rangle\langle\phi_k^T\phi_k\rangle - \langle z_{ik}\rangle^2\langle w_{ik}\rangle^2\langle\phi_k\rangle^T\langle\phi_k\rangle\right)$$

$$+\sum_{k\neq l}\langle z_{ik}\rangle\langle z_{il}\rangle\Delta'_{i,kl}\langle\phi_k\rangle^T\langle\phi_l\rangle$$

In the previous equations, σ_n^{-2} can then be replaced by $\langle\sigma_n^{-2}\rangle = c'/d'$.

Update for σ_w^2: $p(\sigma_w^2|W) \propto p(W|\sigma_w^2)p(\sigma_w^2)$

Given a conjugate, $InvGa(e,f)$ prior, the posterior of σ_w^2 is also inverse-gamma with

$$\begin{aligned} e' &= e + \frac{NK}{2} \qquad (4.23) \\ f' &= f + \frac{1}{2}\sum_{i=1}^{N}\left(\langle w_i\rangle^T\langle w_i\rangle + \operatorname{trace}(\Delta'_i)\right) \qquad (4.24) \end{aligned}$$

4.5.3 Accelerated VB Inference

As with the Dirichlet process, there is a tradeoff in variational inference for the BPFA; the larger K is set, the more accurate the model should be, but the slower the model inference. We here briefly mention a simple remedy for this problem.

Following every iteration, the total factor membership expectations, $\{\langle n_k \rangle\}_{k=1}^{K}$, can be used to assess the relevancy of a particular factor. When this number falls below a small threshold (e.g., 10^{-16}), this factor index can be skipped in following iterations with minimal impact on the convergence of the algorithm. In this way, the algorithm should converge more quickly as the number of iterations increases.

4.5.4 Prediction for New Observations

Given the outputs, $\{\pi, \Phi\}$, the vectors z^* and w^* can be inferred for a new observation, x^*, using a MAP-EM inference algorithm that iterates between z^* and w^*. The equations are similar to those detailed above, with inference for π and Φ removed.

4.6 Experiments

Factor analysis models are useful in many applications, for example, for dimensionality reduction in gene expression analysis [76]. In this section, we demonstrate the performance of the BPFA model on synthetic data, and apply it to the MNIST digits and HGDP-CEPH cell line panel [67] datasets.

4.6.1 A Synthetic Example

For our synthetic example, we generated H from the previously discussed approximation to the Beta process with $a, b = 1$, $K = 100$ and $\phi_k \sim \mathcal{N}(0, I)$ in a $D = 25$ dimensional space. We generated $N = 250$ samples from a Bernoulli process parameterized by H and synthesized X with $W = 1$ and $\sigma_n^2 = 0.0675$. Below, we show results for the model having the highest likelihood selected from five runs, though the results in general were consistent.

In Figure 4.3, we display the ground truth (top) of Z, rearranged for display

purposes. We note that only seven factors were actually used, while several observations contain no factors at all, and thus are pure noise. We initialized our model to $K = 100$ factors, though as the results show (bottom), only a small subset were ultimately used. The inferred $\langle \sigma_n^2 \rangle = 0.0625$ and the elementwise MSE of 0.0186 to the true ΦZ further indicates good performance. For this example, the BPFA model was able to accurately uncover the underlying latent structure of the dataset.

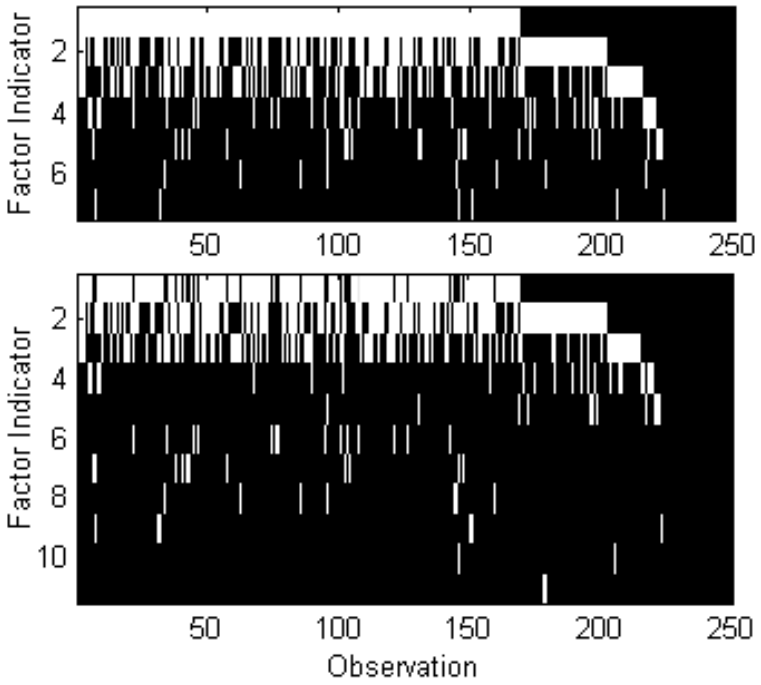


FIGURE 4.3: Synthetic Data: Latent factor indicators, Z, for the true (top) and inferred (bottom) models.

4.6.2 MNIST Handwritten Digits Dataset

We trained our BPFA model on $N = 2500$ odd digits (500 each) from the MNIST digits dataset. Using PCA, we reduced the dimensionality to $D = 350$, which preserved over 99.5% of the total variance within the data. We truncated the BPFA model to $K = 100$ factors initialized using the K-means algorithm and ran five times, selecting the run with the highest likelihood, though again the results were consistent.

In Figure 4.5 below, we show the factor sharing across the digits (left) by calculating the expected number of factors shared between two observations and normalizing by the largest value (0.58); larger boxes indicate more sharing. At right, we show for each of the odd digits the most commonly used factor, followed by the second most used factor *given* the factor to the left. Of particular interest are the digits 3 and 5, where they heavily share the same factor, followed by a factor that differentiates the two numbers.

In Figure 4.4 (top), we plot the sorted values of $\langle \pi \rangle$ inferred by the algorithm. As can be seen, the algorithm inferred a sparse set of factors, fewer than the 100 initially provided. Also in Figure 4.4 (bottom), we show an example of a reconstruction of the number 3 that uses four factors. As can be seen, no single factor can individually approximate the truth as well as their weighted linear combination. We note that the BPFA model was fast, requiring 35 iterations on average to converge and requiring approximately 30 minutes for each run on a 2.66 GHz processor.

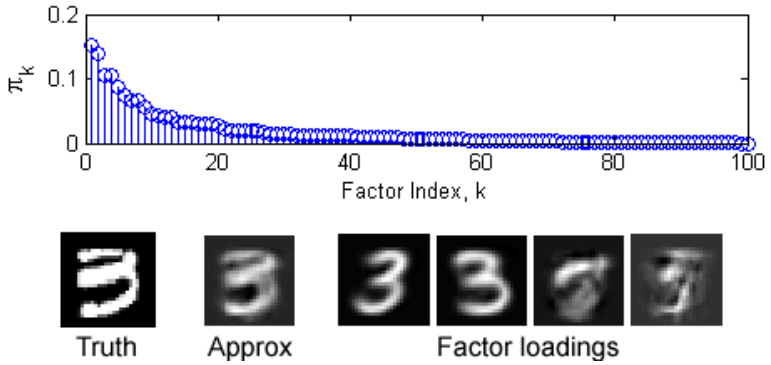


FIGURE 4.4: (top) Inferred π indicating sparse factor usage. (bottom) An example reconstruction.

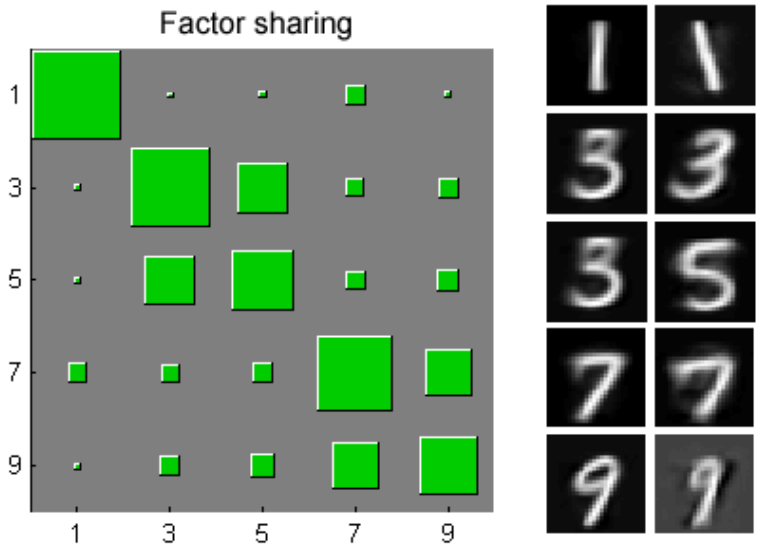


FIGURE 4.5: Left: Expected factor sharing between digits. Right: (left) Most frequently used factors for each digit (right) Most used second factor per digit given left factor.

4.6.3 HGDP-CEPH Cell Line Panel

The HGDP-CEPH Human Genome Diversity Cell Line Panel [67] is a dataset comprising genotypes at $D = 377$ autosomal microsatellite loci sampled from $N = 1056$ individuals in 52 populations across the major geographic regions of the world. It is useful for inferring human evolutionary history and migration.

We ran our model on this dataset initializing $K = 100$ factors, though again, only a subset were significantly used. Figure 4.6 contains the sharing map, as previously calculated for the MNIST dataset, normalized on 0.55. We note the slight differentiation between the Middle East and European regions, a previous issue for this dataset [67].

We also highlight the use of BPFA in denoising. Figure 4.8 shows the original HGDP-CEPH data, as well as the $\Phi(Z \circ W)$ reconstruction projected onto the first 20 principal components of the raw data. The figure shows how the BPFA model was able to substantially reduce the noise level within the data, while still retaining

the essential structure.

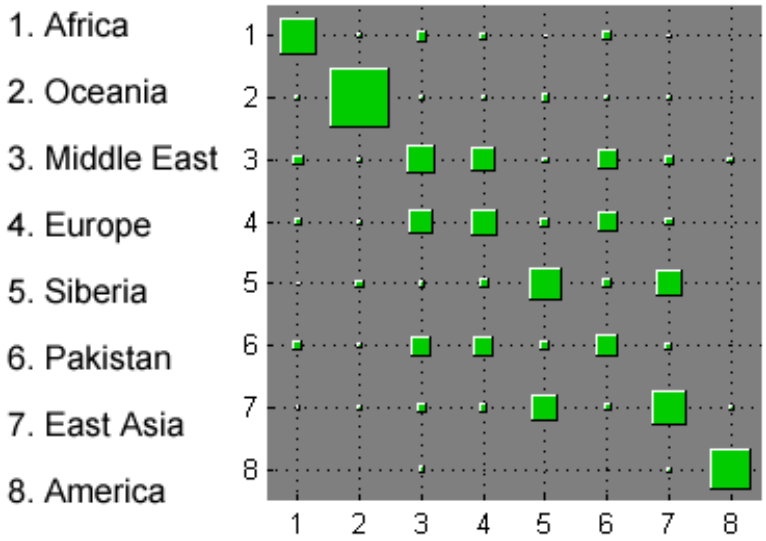


FIGURE 4.6: Factor sharing across geographic regions.

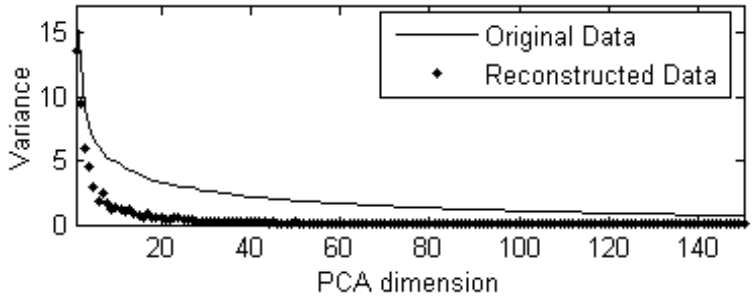


FIGURE 4.7: Variance of HGDP-CEPH data along the first 150 principal components of the raw features for original and reconstructed data.

This is also evident in Figure 4.7, where we plot the variance along these same principal components for the first 150 dimensions. For an apparently noisy dataset such as this, BPFA can potentially be useful as a preprocessing step in conjunction with other algorithms, in this case, for example, the *Structure* [67] or recently proposed *mStruct* [69] algorithms.

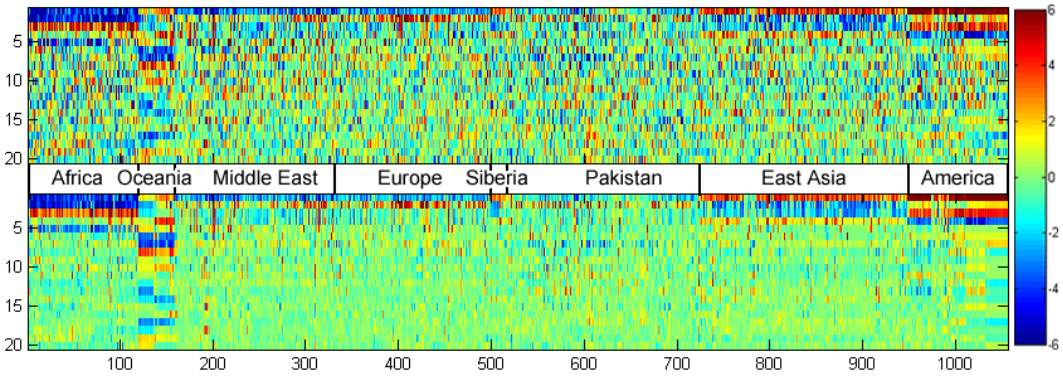


FIGURE 4.8: HGDP-CEPH features projected onto the first 20 principal components of the raw features for the (top) original and (bottom) reconstructed data. The broad geographic breakdown is given between the images.

4.6.4 Learning Dictionaries for Compressive Sensing Applications

We next consider an application of the proposed model to nonparametric dictionary learning. A dictionary, in this case, is simply another name for the factor loading matrix, and is meant to evoke an interpretation of the columns of Φ as a breaking down of the data matrix, X, into a set of fundamental elements (or words) from which all observations are composed. This dictionary also provides a nonorthogonal basis that can be used in compressive sensing [22][16] inversion algorithms. The idea is that, if a good dictionary can be learned on a set of images that are representative of those for which compressive sensing is to be performed, then this dictionary should allow for better signal reconstructions given a fixed number of compressive samples. Before presenting results, we review these ideas in more detail below.

Compressive Sensing and the Relevance Vector Machine

Consider the sparse vector $\theta \in \mathbb{R}^N$, where $\|\theta\|_0 = S$ with $S \ll N$; the vector θ has only S nonzero values and is said to be S-sparse. The motivation behind compressive sensing is that, if the vector $y = \Psi\theta$ is measured instead of θ, where the sensing matrix $\Psi \in \mathbb{R}^{M \times N}$ with $M \ll N$, then θ can be recovered *exactly* with high

probability using an ℓ_1-minimizing inversion algorithm of the form

$$\text{minimize } \|\theta\|_1 \quad \text{subject to} \quad y = \Psi\theta \tag{4.25}$$

For signals with noise, this is relaxed to [74]

$$\text{minimize } \|\theta\|_1 \quad \text{subject to} \quad \|y - \Psi\theta\|_2 < \epsilon \tag{4.26}$$

In practice, the signal is not S-sparse, or even sparse, but is sparse in some basis B. That is, a non-sparse signal, $x \in \mathrm{R}^N$, has a sparse representation, θ, via the transform $x = B\theta$. In this case, the sensed signal is $y = \Psi x$, while sparse inversion is performed for $y = (\Psi B)\theta$. The original signal is then reconstructed by the transform $x = B\theta$ using the learned θ.

Compressive sensing requires that the matrices Ψ and B be *incoherent*, or that

$$\mu(\Psi, B) = \max_{i,j} \sqrt{2}\ \psi_i^T b_j$$

is small, where ψ_i and b_j are basis functions from Ψ and B, respectively. This incoherence – a worst-case measure of how correlated Ψ is with B – is closely linked with the number of measurements, M, that are required to ensure a high probability of perfect reconstruction, and small values are desired [16].

A second important property of the matrix Ψ is that it satisfies the *restricted isometry property* [6], or that

$$(1 - \delta_S)\|\theta\|_2 \leq \|\Psi\theta\|_2 \leq (1 + \delta_S)\|\theta\|_2 \tag{4.27}$$

for $0 < \delta_S < 1$ and all S-sparse vectors θ. This property states that no vector θ is in the null space of Ψ, and smaller values of δ_S indicate that any S columns selected from Ψ at random act nearly as an orthonormal basis for an S-dimensional subspace in $\mathbb{R}^M$. Matrices that satisfy this property with $\delta_{2S} < \sqrt{2} - 1$ guarantee perfect reconstruction of all S-sparse signals, and recovery of the S largest values of all other vectors [16]. Typically, the value of M to be aimed for is $M = 2S$. In this section, we use random matrices, Ψ, where each column is sampled uniformly from the unit sphere in $\mathbb{R}^M$, which satisfies the RIP with high probability [6].

For inversion, we use the relevance vector machine (RVM) [75], also known as *Bayesian compressive sensing* [39], which has the following hierarchical structure.

$$\begin{aligned} y &\sim \mathcal{N}((\Psi B)\theta, \sigma^2 I) \\ \theta &\sim \mathcal{N}\left(0, \text{diag}(\alpha_1^{-1}, \ldots, \alpha_N^{-1})\right) \\ \alpha_n &\overset{iid}{\sim} \text{Gamma}(a, b) \end{aligned} \tag{4.28}$$

where a and b are small numbers, (e.g., 10^{-6}). We use variational Bayes to find a local optimal solution to the model [8][77]. In this case, the RVM can be viewed as an iteratively updated ridge regression solution [35], where each element θ_n has a unique penalty term, α_n. The gamma prior encourages each $\alpha_n \to \infty$, which thereby encourages $\theta_n \to 0$. Therefore, the RVM produces sparse solutions for θ by shrinking all unnecessary coefficients out of the model, though this does not necessarily produce the minimum ℓ_1 solution. Nevertheless, the RVM performs competitively with other inversion algorithms, including ℓ_1 minimization [82].

Basis Design and Compressive Sensing with BPFA

The purpose of the BPFA model in this scenario is to learn a specific basis, B, for a set of signals of interest. To do this, we assume access to a set of images, $\mathcal{I} = \{I_r\}_{r=1}^R$, from which are extracted a large set of square patches, $\mathcal{P} = \{P_t\}_{t=1}^T$. Specifically, we consider $R = 192$ grayscale images from the Yale Faces database,[1] where each $I_r \in \mathbb{R}^{128\times 96}$. We extracted 100 patches from each image of size $P_t \in \mathbb{R}^{8\times 8}$ selected randomly from within the image. These patches were then stretched into vectors, $x \in \mathbb{R}^{64}$, producing the data matrix $X \in \mathbb{R}^{64\times 13200}$. The BPFA model was then run with $K = 200$ initialized factors and a dictionary matrix (i.e., factor loading matrix), Φ, was learned that ultimately required 81 of the original 200 factors. Therefore, each patch in an image is a sparse linear combination of 81 possible 8×8 elementary patches.

[1] http://cvc.yale.edu/projects/yalefaces/yalefaces.html

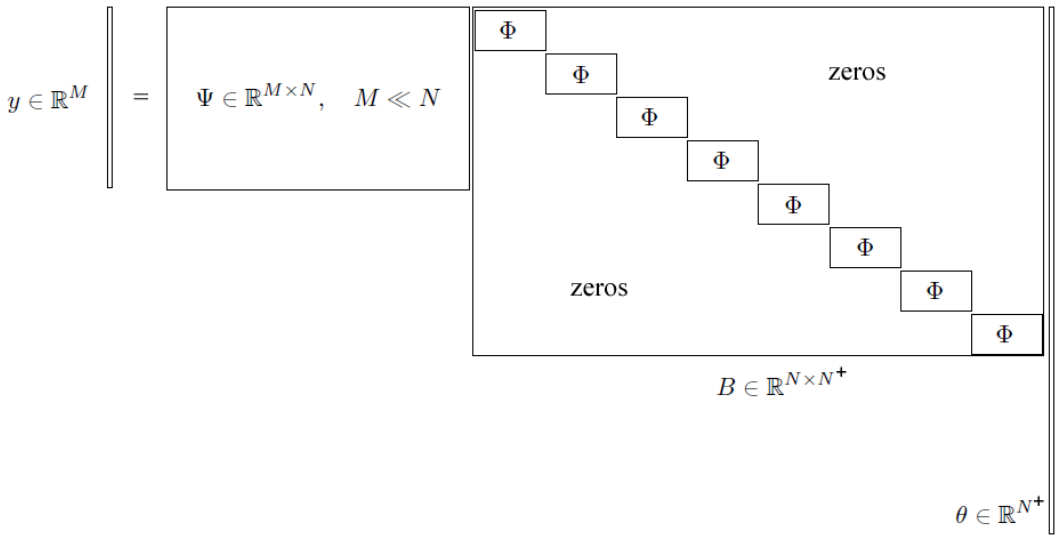


FIGURE 4.9: An illustration of the constructed basis using the learned dictionary Φ. Each block diagonal matrix, $\Phi \in \mathbb{R}^{64 \times 81}$, is responsible for reconstructing a patch in the original image and each column is normalized prior to inversion. For our application, we broke each image into non-overlapping 8×8 patches for reconstruction. The number of sparse coefficients, θ, to be learned therefore increases to $N^+ = 1.265625N$.

Once Φ is learned, all other parameters are thrown away, and Φ is used to construct a basis as shown in Figure 4.9. For reconstruction, a test image is broken into 8×8 contiguous blocks and the inversion proceeds as pictured. Each block-diagonal dictionary matrix is responsible for reconstructing one patch in an images using the corresponding coefficients in the sparse vector θ. We observe that, due to the overcomplete nature of the dictionary, the number of sparse coefficients in θ to be learned actually increases. However, since the model attempts to shrink each coefficient to zero, this does not lead to performance degradation, or an increase in the number of nonzero coefficients, as will be seen.

As a comparison, we consider three other basis options. These include two commonly used bases, the 2-dimensional DCT and wavelet bases [2][46], as well as the principal component (PCA) basis [42] learned from the matrix X. This PCA basis replaces the dictionary Φ during inversion. We note that the learned dictionary, though normalized beforehand, does not form an orthonormal basis within each block

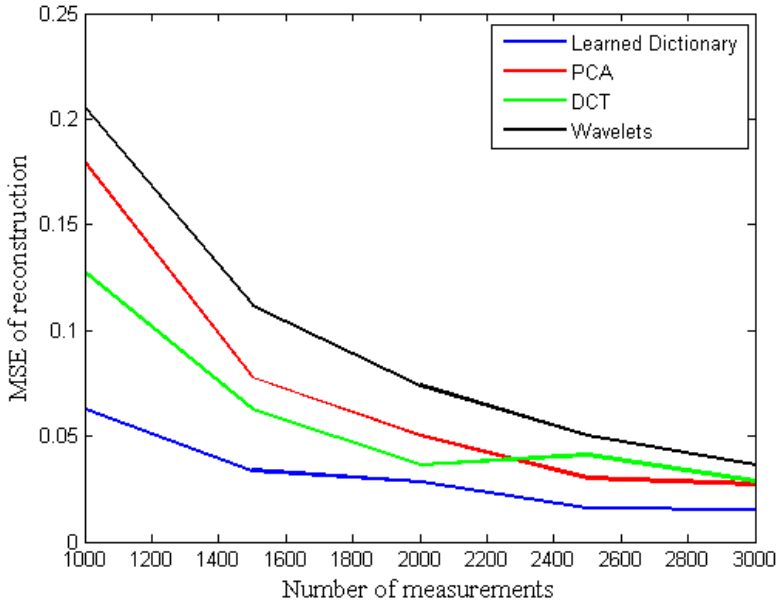


FIGURE 4.10: The reconstructed MSE for different basis representations and different compressive measurement numbers.

of B.

In Figure 4.11, we show reconstruction results using a test image for several different numbers of compressive measurements. The reconstruction error of these images, measured as the MSE of the reconstruction to the original image, is shown in Figure 4.10, where we see that the learned basis outperforms all other bases. We show two of these basis, the BPFA-learned dictionary and the PCA dictionary in Figure 4.12. The dictionary elements of the PCA dictionary explain the checkerboard pattern in many reconstructions. The relaxing of the orthogonality constraint in the BPFA algorithm results in dictionary elements that are more natural for characterizing the images under consideration, which is a result of learning the dictionary elements on a subset of these images of interest. In Figure 4.13, we show the sparsely learned coefficients sorted by absolute value. We see that, despite the increase in the number of coefficients to be learned using the overcomplete BPFA dictionary, the sparse usage of this dictionary is comparable with other sparse basis representations.

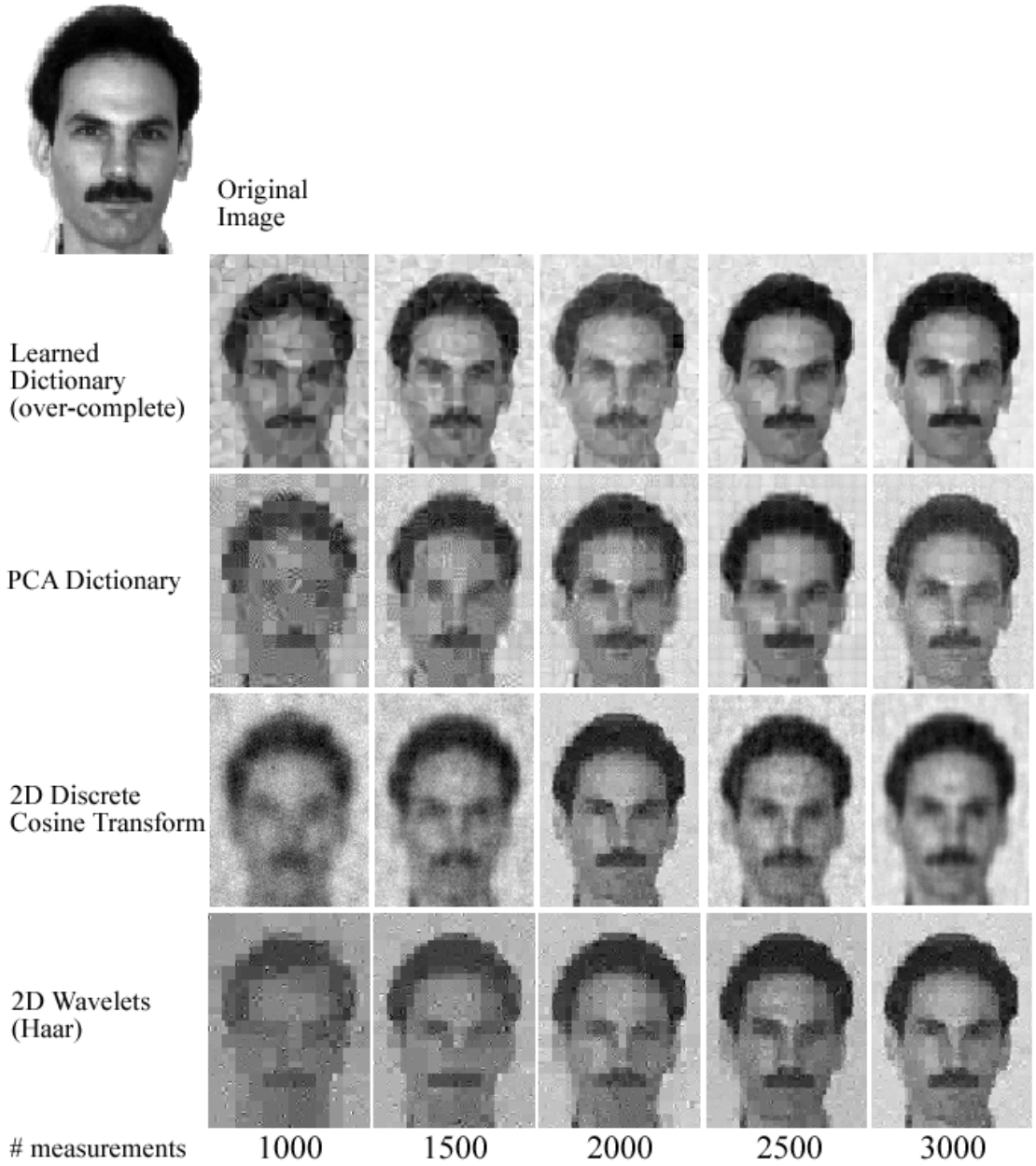


FIGURE 4.11: Compressive sensing reconstruction results using the RVM for the dictionary basis learned with BPFA, the PCA basis and the 2D DCT and wavelet bases for different numbers of compressive measurements.

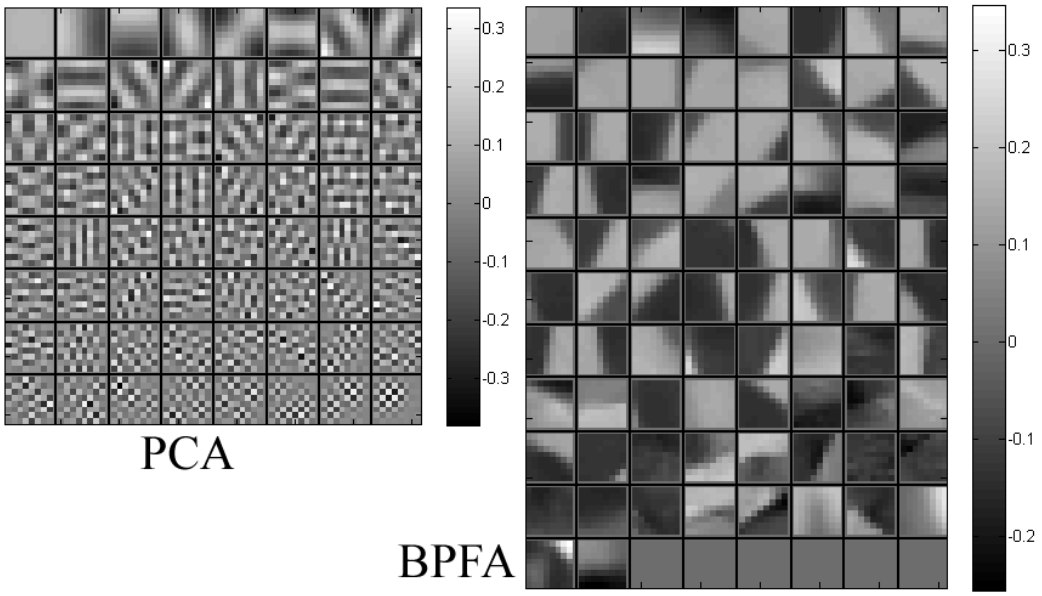


FIGURE 4.12: The PCA dictionary (left) and BPFA dictionary (right) used for inversion. The relaxation of the orthogonality constraint for BPFA can be seen to produce dictionaries that are more natural for reconstructing the images of interest.

4.7 Conclusion

We have presented a *beta process factor analysis* (BPFA) model for performing nonparametric factor analysis with a potentially infinite number of factors. As with the Dirichlet process prior used for mixture modeling, the beta process is a fully Bayesian prior that assures the sharing of a sparse subset of factors among all observations. Taking advantage of conjugacy within the model, a variational Bayes algorithm was developed for fast model inference requiring an approximation comparable to the finite Dirichlet distribution's approximation to the infinite Dirichlet process. Results were shown on synthetic data, as well as the MNIST handwritten digits and HGDP-CEPH cell line panel datasets. We have also shown an application of the BPFA model to dictionary learning for designing a basis that can be used in compressive sensing inversion algorithms. This learned basis produced better reconstruction results than other "off-the-shelf" bases.

While several nonparametric factor analysis models have been proposed for ap-

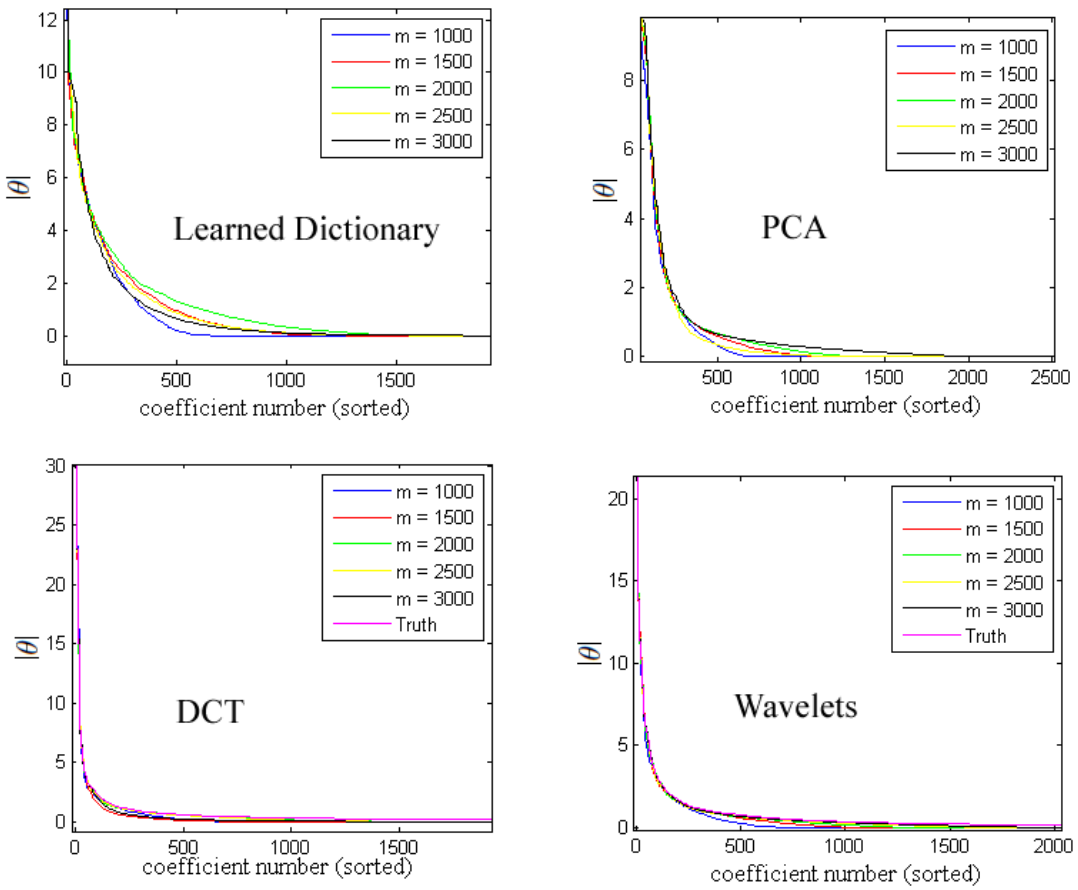


FIGURE 4.13: The learned, sparse coefficients sorted by absolute value. The overcomplete dictionary required inference for 15,744 coefficients, compared with 12,288 coefficients for the other bases. However, the inferred sparseness is comparable.

plications such as independent components analysis [44] and gene expression analysis [62, 47], these models rely on the Indian buffet process and therefore do not have fast variational solutions - an intended contribution of this chapter. Furthermore, while the formal link has been made between the IBF and the beta process [73], we believe our further development and application to factor analysis to be novel.

5

A Stick-Breaking Construction of the Beta Process

5.1 Abstract

In this chapter, we present and derive a new stick-breaking construction of the beta process [57]. The construction is closely related to a special case of the stick-breaking construction of the Dirichlet process [68] applied to the beta distribution. We derive an inference procedure that relies on Monte Carlo integration to reduce the number of parameters to be inferred, and present results on synthetic data, the MNIST handwritten digits data set and a time-evolving gene expression data set.

5.2 Introduction

The Dirichlet process [27] is a powerful Bayesian nonparametric prior for mixture models. There are two principle methods for drawing from this infinite-dimensional prior: (*i*) the Chinese restaurant process [11], in which samples are drawn from a marginalized Dirichlet process and implicitly construct the prior; and (*ii*) the stick-breaking process [68], which is a fully Bayesian construction of the Dirichlet process.

Similarly, the beta process [34] is receiving significant use recently as a nonpara-

metric prior for latent factor models [30, 73]. This infinite-dimensional prior can be drawn via marginalization using the Indian buffet process [31], where samples again construct the prior. However, unlike the Dirichlet process, the fully Bayesian stick-breaking construction of the beta process has yet to be derived (though related methods exist [73, 72], reviewed in Section 2).

To review, a Dirichlet process, G, can be constructed according to the following stick-breaking process [68, 36],

$$\begin{aligned} G &= \sum_{i=1}^{\infty} V_i \prod_{j=1}^{i-1}(1-V_j)\delta_{\theta_i} \\ V_i &\overset{iid}{\sim} \text{Beta}(1,\alpha) \\ \theta_i &\overset{iid}{\sim} G_0 \end{aligned} \tag{5.1}$$

This stick-breaking process is so-called because proportions, V_i, are sequentially broken from the remaining length, $\prod_{j=1}^{i-1}(1-V_j)$, of a unit-length stick. This produces a probability (or weight), $V_i \prod_{j=1}^{i-1}(1-V_j)$, that can be visually represented as one of an infinite number of contiguous sections cut out of a unit-length stick. As i increases, these weights stochastically decrease, since smaller and smaller fractions of the stick remain, and so only a small number of the infinite number of weights have appreciable value. By construction, these weights occur first, which allows for practical implementation of this prior.

The contribution of this chapter is the derivation of a stick-breaking construction of the beta process. We use a little-known property of the constructive definition in [68], which is equally applicable to the beta distribution – a two-dimensional Dirichlet distribution. The construction presented here will be seen to result from an infinite collection of these stick-breaking constructions of the beta distribution.

This chapter is organized as follows. In Section 5.3, we review the beta process,

the stick-breaking construction of the beta distribution, as well as related work in this area. In Section 5.5, we present the stick-breaking construction of the beta process and its derivation. We derive an inference procedure for the construction in Section 5.4 and present experimental results on synthetic data, the MNIST handwritten digits and gene expression data in Section 5.6.

5.3 The Beta Process

Let H_0 be a continuous measure on the space $(\Theta, \mathcal{B})$ and let $H_0(\Theta) = \gamma$. Also, let α be a positive scalar and define the process H_K as follows,

$$\begin{aligned} H_K &= \sum_{k=1}^{K} \pi_k \delta_{\theta_k} \\ \pi_k &\overset{iid}{\sim} \text{Beta}\left(\frac{\alpha\gamma}{K}, \alpha(1 - \frac{\gamma}{K})\right) \\ \theta_k &\overset{iid}{\sim} \frac{1}{\gamma} H_0 \end{aligned} \tag{5.2}$$

then as $K \to \infty$, $H_K \to H$ and H is a beta process, which we denote $H \sim \text{BP}(\alpha H_0)$.

We avoid a complete measure-theoretic definition, since the stick-breaking construction to be presented is derived in reference to the limit of (5.2). That H is a beta process can be shown in the following way: Integrating out $\underline{\pi}^{(K)} = (\pi_1, \ldots, \pi_K)^T \in (0,1)^K$, letting $K \to \infty$ and sampling from this marginal distribution produces the two-parameter extension of the Indian buffet process discussed in [73], which is shown to have the beta process as its underlying de Finetti mixing distribution. We also observe that a different parametrization of the beta process is used in this chapter. This has the advantage of a cleaner derivation, but we also note that this different parametrization arises from a slightly different definition, where the measure H_0 is not necessarily a probability measure. The second parameter in this case comes from the setting of $H_0(\Theta)$, i.e., from a scaling of the H_0 defined in the previous chapter.

These two parameterizations can be made equivalent via the setting of a, b, α and γ.

Before deriving the stick-breaking construction of the beta process, we review a property of the beta *distribution* that will be central to the construction. We also review related work to distinguish the presented construction from other constructions in the literature.

5.3.1 A Construction of the Beta Distribution

The constructive definition of a Dirichlet prior derived in [68] applies to more than the infinite-dimensional Dirichlet process. In fact, it is applicable to Dirichlet priors of any dimension, of which the beta distribution can be viewed as a special, two-dimensional case. [1] Focusing on this special case, Sethuraman showed that one can sample

$$\pi \sim \text{Beta}(a, b) \tag{5.3}$$

according to the following stick-breaking construction,

$$\begin{aligned} \pi &= \sum_{i=1}^{\infty} V_i \prod_{j=1}^{i-1} (1 - V_j) \mathbb{I}(Y_i = 1) \\ V_i &\overset{iid}{\sim} \text{Beta}(1, a + b) \\ Y_i &\overset{iid}{\sim} \text{Bernoulli}\left(\frac{a}{a+b}\right) \end{aligned} \tag{5.4}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function.

In this construction, weights are drawn according to the standard stick-breaking construction of the DP [36], as well as their respective locations, which are independent of the weights and iid among themselves. The major difference is that the set of locations is finite, 0 or 1, which results in more than one term being active in the summation.

[1] We thank Jayaram Sethuraman for his valuable correspondence regarding his constructive definition.

The proof of this construction is in Chapter 1.4, with $g_0 := [\gamma/K(1-\gamma/K)]$. Sethuraman also proves this implicitly in the following way: Using notation from [68], let the space, $\mathcal{X} = \{0, 1\}$, and the prior measure, α, be $\alpha(1) = a$, $\alpha(0) = b$, and therefore $\alpha(\mathcal{X}) = a + b$. Carrying out the proof in [68] for this particular space and measure yields (5.4). We note that this α is different from that in (5.2).

5.3.2 Related Work

To our knowledge, there are currently three related constructions, each of which differs significantly from that presented here. The first construction, proposed by [71], is presented specifically for the Indian buffet process (IBP) prior. The fully Bayesian generative process from which the IBP and this construction are derived replaces the beta distribution in (5.2) with $\text{Beta}(\frac{\alpha}{K}, 1)$. This small change greatly facilitates this construction, since the parameter 1 in $\text{Beta}(\frac{\alpha}{K}, 1)$ allows for a necessary simplification of the beta distribution. This construction does not extend to the two-parameter generalization of the IBP [30], which is equivalent in the infinite limit to the marginalized representation in (5.2).

A second method for drawing directly from the beta process prior has been presented in [73], and more recently in [72] as a special case of a more general power-law representation of the IBP. In this representation, no stick-breaking takes place of the form in (5.1), but rather the weight for each location is simply beta-distributed, as opposed to the usual function of multiple beta-distributed random variables. The derivation relies heavily upon connecting the marginalized process to the fully Bayesian representation, which does not factor into the similar derivation for the DP [68]. This of course does not detract from the result, which appears to have a simpler inference procedure than that presented here.

A third representation presented in [72] is based on the inverse Lévy method [79] and exists in theory only. The derivation of this representation requires significant

prior knowledge regarding Lévy processes and does not simplify to an analytic stick-breaking form.

5.4 A Stick-Breaking Construction of the Beta Process

We now define and briefly discuss the stick-breaking construction of the beta process, followed by its derivation. Let α and H_0 be defined as in (5.2). If H is constructed according to the following process,

$$\begin{aligned} H &= \sum_{i=1}^{\infty} \sum_{j=1}^{C_i} V_{ij}^{(i)} \prod_{\ell=1}^{i-1} (1 - V_{ij}^{(\ell)}) \delta_{\theta_{ij}} \\ C_i &\overset{iid}{\sim} \text{Poisson}(\gamma) \\ V_{ij}^{(\ell)} &\overset{iid}{\sim} \text{Beta}(1, \alpha) \\ \theta_{ij} &\overset{iid}{\sim} \frac{1}{\gamma} H_0 \end{aligned} \tag{5.5}$$

then $H \sim \text{BP}(\alpha H_0)$.

Since the first row of (5.5) may be unclear at first sight, we expand it for the first few values of i below,

$$\begin{aligned} H &= \sum_{j=1}^{C_1} V_{1,j}^{(1)} \delta_{\theta_{1,j}} + \\ &\quad \sum_{j=1}^{C_2} V_{2,j}^{(2)} (1 - V_{2,j}^{(1)}) \delta_{\theta_{2,j}} + \\ &\quad \sum_{j=1}^{C_3} V_{3,j}^{(3)} (1 - V_{3,j}^{(2)})(1 - V_{3,j}^{(1)}) \delta_{\theta_{3,j}} + \cdots \end{aligned} \tag{5.6}$$

For each value of i, which we refer to as a "round," there are C_i atoms, where C_i is itself random and drawn from Poisson(γ). Therefore, every atom is defined by

two subscripts, (i, j). The mass associated with each atom in round i is equal to the i^{th} break from an *atom-specific* stick, where the stick-breaking weights follow a Beta$(1, \alpha)$ stick-breaking process (as in (5.1)). Superscripts are used to index the i random variables that construct the weight on atom θ_{ij}. Since the number of breaks from the unit-length stick prior to obtaining a weight increases with each level in (5.6), the weights stochastically decrease as i increases, in a similar manner as in the stick-breaking construction of the Dirichlet process (5.1).

Since the expectation of the mass on the k^{th} atom drawn overall does not simplify to a compact and transparent form, we omit its presentation here. However, we note the following relationship between α and γ in the construction. As α decreases, weights decay more rapidly as i increases, since smaller fractions of each unit-length stick remains prior to obtaining a weight. As α increases, the weights decay more gradually over several rounds. The expected weight on an atom in round i is equal to $\alpha^{(i-1)}/(1+\alpha)^i$. The number of atoms in each round is controlled by γ.

5.4.1 Derivation of the Construction

Starting with (5.2), we now show how Sethuraman's constructive definition of the beta distribution can be used to derive that the infinite limit of (5.2) has (5.5) as an alternate representation that is equal in distribution. We begin by observing that, according to (5.4), each π_k value can be drawn as follows,

$$\begin{aligned} \pi_k &= \sum_{l=1}^{\infty} \hat{V}_{kl} \prod_{m=1}^{l-1} (1 - \hat{V}_{km}) \mathbb{I}(\hat{Y}_{kl} = 1) \\ \hat{V}_{kl} &\overset{iid}{\sim} \text{Beta}(1, \alpha) \\ \hat{Y}_{kl} &\overset{iid}{\sim} \text{Bernoulli}\left(\frac{\gamma}{K}\right) \end{aligned} \tag{5.7}$$

where the marker $\hat{}$ is introduced because V will later be re-indexed values of $\hat{V}$. We also make the observation that, if the sum is instead taken to K', and we then let

$K' \to \infty$, then this truncated representation converges to (5.7).

This suggests the following procedure for constructing the limit of the vector $\underline{\pi}^{(K)}$ in (5.2). We define the matrices $\hat{\underline{V}} \in (0,1)^{K \times K}$ and $\hat{\underline{Y}} \in \{0,1\}^{K \times K}$, where

$$\begin{aligned} \hat{V}_{kl} &\overset{iid}{\sim} \text{Beta}(1, \alpha) \\ \hat{Y}_{kl} &\overset{iid}{\sim} \text{Bernoulli}\left(\frac{\gamma}{K}\right) \end{aligned} \tag{5.8}$$

for $k = 1, \ldots, K$ and $l = 1, \ldots, K$. The K-truncated weight, π_k, is then constructed "horizontally" by looking at the k^{th} row of $\hat{\underline{V}}$ and $\hat{\underline{Y}}$, and where we define that the error of the truncation is assigned to $1 - \pi_k$ (i.e., $Y_{k,l'} := 0$ for the extension $l' > K$.)

It can be seen from the matrix definitions in (5.8) and the underlying function of these two matrices, defined for each row as a K-truncated version of (5.7), that in the limit as $K \to \infty$, this representation converges to the infinite beta process when viewed vertically, and to a construction of the individual beta-distributed random variables when viewed horizontally, each of which occur simultaneously.

Before using these two matrices to derive (5.5), we derive a probability that will be used in the infinite limit. For a given column, i, of (5.8), we calculate the probability that, for a particular row, k, there is at least one $\hat{Y} = 1$ in the set $\{\hat{Y}_{k,1}, \ldots, \hat{Y}_{k,i-1}\}$, in other words, the probability that $\sum_{i'=1}^{i-1} \hat{Y}_{ki'} > 0$. This value is

$$\mathbb{P}\left(\sum_{i'=1}^{i-1} \hat{Y}_{ki'} > 0 | \gamma, K\right) = 1 - (1 - \frac{\gamma}{K})^{i-1} \tag{5.9}$$

In the limit as $K \to \infty$, this can be shown to converge to zero for all fixed values of i.

As with the Dirichlet process, the problem with drawing each π_k explicitly in the limit of (5.2) is that there are an infinite number of them, and any given π_k is equal to zero with probability one. With the representation in (5.8), this problem appears to have doubled, since there are now an infinite number of random variables

to sample in two dimensions, rather than one. However, this is only true when viewed *horizontally*. When viewed vertically, drawing the values of interest becomes manageable.

First, we observe that, in (5.8), we only care about the set of indices $\{(k,l) : \hat{Y}_{kl} = 1\}$, since these are the locations which indicate that mass is to be added to their respective π_k values. Therefore, we seek to bypass the drawing of all indices for which $\hat{Y} = 0$, and directly draw those indices for which $\hat{Y} = 1$.

To do this, we use a property of the binomial distribution. For any column, i, of $\hat{\underline{Y}}$, the number of nonzero locations, $\sum_{k=1}^{K} \hat{Y}_{ki}$, has the Binomial$(K, \frac{\gamma}{K})$ distribution. Also, it is well-known that

$$\text{Poisson}(\gamma) = \lim_{K\to\infty} \text{Binomial}\left(K, \frac{\gamma}{K}\right) \tag{5.10}$$

Therefore, in the limit as $K \to \infty$, the sum of each column (as well as row) of $\hat{\underline{Y}}$ produces a random variable with a Poisson(γ) distribution. This suggests the procedure of first drawing the number of nonzero locations for each column, followed by their corresponding indices.

Returning to (5.8), given the number of nonzero locations in column i, $\sum_{k=1}^{K} \hat{Y}_{ki} \sim$ Binomial$(K, \frac{\gamma}{K})$, finding the indices of these locations then becomes a process of sampling uniformly from $\{1, \ldots, K\}$ without replacement. Moreover, since there is a one-to-one correspondence between these indices and the atoms, $\theta_1, \ldots, \theta_K \overset{iid}{\sim} \frac{1}{\gamma} H_0$, which they index, this is equivalent to selecting from the set of atoms, $\{\theta_1, \ldots, \theta_K\}$, uniformly without replacement.

A third more conceptual process, which will aid the derivation, is as follows: Sample the $\sum_{k=1}^{K} \hat{Y}_{ki}$ nonzero indices for column i one at a time. After an index, k', is obtained, check $\{\hat{Y}_{k',1}, \ldots, \hat{Y}_{k',i-1}\}$ to see whether this index has already been drawn. If it has, add the corresponding mass, $V_{k'i} \prod_{l=1}^{i-1}(1 - V_{k'l})$, to the tally for $\pi_{k'}$. If it has not, draw a new atom, $\theta_{k'} \sim \frac{1}{\gamma} H_0$, and associate the mass with this atom.

The derivation concludes by observing the behavior of this last process as $K \to \infty$. We first reiterate that, in the limit as $K \to \infty$, the count of nonzero locations for each column is independent and identically distributed as Poisson(γ). Therefore, for $i = 1, 2, \ldots$, we can draw these numbers, $C_i := \sum_{k=1}^{\infty} \hat{Y}_{ki}$, as

$$C_i \overset{iid}{\sim} \text{Poisson}(\gamma) \tag{5.11}$$

We next need to sample index values uniformly from the positive integers, $\mathbb{N}$. However, we recall from (5.9) that for all fixed values of i, the probability that the drawn index will have previously seen a one is equal to zero. Therefore, using the conceptual process defined above, we can bypass sampling the index value and directly sample the atom which it indexes. Also, we note that the "without replacement" constraint no longer factors.

The final step is simply a matter of re-indexing. Let the function $\sigma_i(j)$ map the input $j \in \{1, \ldots, C_i\}$ to the index of the j^{th} nonzero element drawn in column i, as discussed above. Then the re-indexed random variables $V_{ij}^{(i)} := \hat{V}_{\sigma_i(j),i}$ and $V_{ij}^{(\ell)} := \hat{V}_{\sigma_i(j),\ell}$, where $\ell < i$. We similarly re-index $\theta_{\sigma_i(j)}$ as $\theta_{ij} := \theta_{\sigma_i(j)}$, letting the double and single subscripts remove ambiguity, and hence no $\hat{}$ marker is used. The addition of a subscript/superscript in the two cases above arises from ordering the nonzero locations for each column of (5.8), i.e., the original index values for the selected rows of each column are being mapped to $1, 2, \ldots$ separately for each column in a many-to-one manner. The result of this re-indexing is the process given in (5.5).

5.5 Inference for the Stick-Breaking Construction

For inference, we integrate out all stick-breaking random variables, V, using Monte Carlo integration [29], which significantly reduces the number of random variables to be learned. As a second aid for inference, we introduce the latent round-indicator

variable,

$$d_k := 1 + \sum_{i=1}^{\infty} \mathbb{I}\left(\sum_{j=1}^{i} C_j < k\right) \tag{5.12}$$

The equality $d_k = i$ indicates that the k^{th} atom drawn overall occurred in round i. Note that, given $\{d_k\}_{k=1}^{\infty}$, we can reconstruct $\{C_i\}_{i=1}^{\infty}$. Given these latent indicators, the generative process is rewritten as,

$$\begin{aligned} H \mid \{d_k\}_{k=1}^{\infty} &= \sum_{k=1}^{\infty} V_{k,d_k} \prod_{j=1}^{d_k-1} (1 - V_{kj}) \delta_{\theta_k} \\ V_{kj} &\overset{iid}{\sim} \text{Beta}(1, \alpha) \\ \theta_k &\overset{iid}{\sim} \frac{1}{\gamma} H_0 \end{aligned} \tag{5.13}$$

where, for clarity in what follows, we've avoided introducing a third marker (e.g., $\tilde{V}$) after this re-indexing.

Data is generated iid from H via a Bernoulli process and take the form of infinite-dimensional binary vectors, $z_n \in \{0, 1\}^{\infty}$, where

$$z_{nk} \sim \text{Bernoulli}\left(V_{k,d_k} \prod_{j=1}^{d_k-1} (1 - V_{kj})\right) \tag{5.14}$$

The sufficient statistics calculated from $\{z_n\}_{n=1}^{N}$ are the counts along each dimension, k,

$$m_{1k} = \sum_{n=1}^{N} \mathbb{I}(z_{nk} = 1), \; m_{0k} = \sum_{n=1}^{N} \mathbb{I}(z_{nk} = 0) \tag{5.15}$$

5.5.1 Inference for d_k

With each iteration, we sample the sequence $\{d_k\}_{k=1}^{K}$ without using future values from the previous iteration; the value of K is random and equals the number of nonzero

m_{1k}. The probability that the k^{th} atom was observed in round i is proportional to

$$p\left(d_k = i | \{d_l\}_{l=1}^{k-1}, \{z_{nk}\}_{n=1}^{N}, \alpha, \gamma\right) \propto p(\{z_{nk}\}_{n=1}^{N} | d_k = i, \alpha) p(d_k = i | \{d_l\}_{l=1}^{k-1}, \gamma) \quad (5.16)$$

Below, we discuss the likelihood and prior terms, followed by an approximation to the posterior.

Likelihood Term

The integral to be solved for integrating out the random variables $\{V_{kj}\}_{j=1}^{i}$ is

$$p(\{z_{nk}\}_{n=1}^{N} | d_k = i, \alpha) = \int_{(0,1)^i} f(\{V_{kj}\}_1^i)^{m_{1k}} \{1 - f(\{V_{kj}\}_1^i)\}^{m_{0k}} p(\{V_{kj}\}_1^i | \alpha) \, d\vec{V} \quad (5.17)$$

where $f(\cdot)$ is the stick-breaking function used in (5.14). Though this integral can be analytically solved for integer values of m_{0k} via the binomial expansion, we have found that the resulting sum of terms leads to computational precision issues for even small sample sizes. Therefore, we use Monte Carlo methods to approximate this integral.

For $s = 1, \ldots, S$ samples, $\{V_{kj}^{(s)}\}_{j=1}^{i}$, drawn iid from Beta$(1, \alpha)$, we calculate

$$p(\{z_{nk}\}_{n=1}^{N} | d_k = i, \alpha) \approx \frac{1}{S} \sum_{s=1}^{S} f(\{V_{kj}^{(s)}\}_{j=1}^{i})^{m_{1k}} \{1 - f(\{V_{kj}^{(s)}\}_{j=1}^{i})\}^{m_{0k}} \quad (5.18)$$

This approximation allows for the use of natural logarithms in calculating the posterior, which was not possible with the analytic solution. Also, to reduce computations, we note that at most two random variables need to be drawn to perform the above stick-breaking, one random variable for the proportion and one for the error; this is detailed in the appendix.

Prior Term

The prior for the sequence of indicators $d_1, d_2, \ldots$ is the equivalent sequential process for sampling $C_1, C_2, \ldots$, where $C_i = \sum_{k=1}^{\infty} \mathbb{I}(d_k = i) \sim$ Poisson(γ). Let $\#_{d_{k-1}} =$

$\sum_{j=1}^{k-1} \mathbb{I}(d_j = d_{k-1})$ and let $\mathbb{P}_\gamma(\cdot)$ denote the Poisson distribution with parameter γ. Then it can be shown that

$$p(d_k = d_{k-1}|\gamma, \#_{d_{k-1}}) = \frac{\mathbb{P}_\gamma(C > \#_{d_{k-1}})}{\mathbb{P}_\gamma(C \geq \#_{d_{k-1}})} \tag{5.19}$$

Also, for $h = 1, 2, \ldots$, the probability

$$p(d_k = d_{k-1} + h|\gamma, \#_{d_{k-1}}) = \tag{5.20}$$

$$\left(1 - \frac{\mathbb{P}_\gamma(C > \#_{d_{k-1}})}{\mathbb{P}_\gamma(C \geq \#_{d_{k-1}})}\right) \mathbb{P}_\gamma(C > 0)\mathbb{P}_\gamma(C = 0)^{h-1}$$

Since $d_k \not< d_{k-1}$, these two terms complete the prior.

Posterior of d_k

For the posterior, the normalizing constant requires integration over $h = 0, 1, 2, \ldots$, which is not possible given the proposed sampling method. We therefore propose incrementing h until the resulting truncated probability of the largest value of h falls below a threshold (e.g., 10^{-6}). We have found that the probabilities tend to decrease rapidly for $h > 1$.

5.5.2 Inference for γ

Given $d_1, d_2, \ldots$, the values $C_1, C_2, \ldots$ can be reconstructed and a posterior for γ can be obtained using a conjugate gamma prior. Since the value of d_K may not be the last in the sequence composing C_{d_K}, this value can be "completed" by sampling from the prior, which can additionally serve as proposal factors.

5.5.3 *Inference for α*

Using (5.18), we again integrate out all stick-breaking random variables to calculate the posterior of α,

$$p(\alpha|\{z_n\}_1^N, \{d_k\}_1^K) \propto \prod_{k=1}^{K} p(\{z_{nk}\}_1^N|\alpha, \{d_k\}_1^K)p(\alpha)$$

Since this is not possible for the positive, real-valued α, we approximate this posterior by discretizing the space. Specifically, using the value of α from the previous iteration, α_{prev}, we perform Monte Carlo integration at the points $\{\alpha_{\text{prev}} + t\Delta\alpha\}_{t=-T}^{T}$, ensuring that $\alpha_{\text{prev}} - T\Delta\alpha > 0$. We use an improper, uniform prior for α, with the resulting probability therefore being the normalized likelihood over the discrete set of selected points. As with sampling d_k, we again extend the limits beyond $\alpha_{\text{prev}} \pm T\Delta\alpha$, checking that the tails of the resulting probability fall below a threshold.

5.5.4 *Inference for $p(z_{nk} = 1|\alpha, d_k, Z_{\text{prev}})$*

In latent factor models, [31], the vectors $\{z_n\}_{n=1}^N$ are to be learned with the rest of the model parameters. To calculate the posterior of a given binary indicator therefore requires a prior, which we calculate as follows

$$\begin{aligned} p(z_{nk} = 1|\alpha, d_k, Z_{\text{prev}}) &= \int_{(0,1)^{d_k}} p(z_{nk} = 1|\vec{V})p(\vec{V}|\alpha, d_k, Z_{\text{prev}})\ d\vec{V} \qquad (5.21) \\ &= \frac{\int_{(0,1)^{d_k}} p(z_{nk} = 1|\vec{V})p(Z_{\text{prev}}|\vec{V})p(\vec{V}|\alpha, d_k)\ d\vec{V}}{\int_{(0,1)^{d_k}} p(Z_{\text{prev}}|\vec{V})p(\vec{V}|\alpha, d_k)\ d\vec{V}} \end{aligned}$$

We again perform Monte Carlo integration (5.18), where the numerator increments the count m_{1k} of the denominator by one. For computational speed, we treat the previous latent indicators, Z_{prev}, as a block [36], allowing this probability to remain fixed when sampling the new matrix, Z.

5.6 Experiments

We present experimental results on three data sets: (*i*) A synthetic data set; (*ii*) the MNIST handwritten digits data set (digits 3, 5 and 8); and (*iii*) a time-evolving gene expression data set.

5.6.1 Synthetic Data

For the synthetic problem, we investigate the ability of the inference procedure in Section 5.5 to learn the underlying α and γ used in generating H. We use the representation in (5.2) to generate $\underline{\pi}^{(K)}$ for $K = 100{,}000$. This provides a sample of $\underline{\pi}^{(K)}$ that approximates the infinite beta process well for smaller values of α and γ. We then sample $\{z_n\}_{n=1}^{1000}$ from a Bernoulli process and remove all dimensions, k, for which $m_{1k} = 0$. Since the weights in (5.13) are stochastically decreasing as k increases, while the representation in (5.2) is exchangeable in k, we reorder the dimensions of $\{z_n\}_{n=1}^{1000}$ so that $m_{1,1} \geq m_{1,2} \geq \ldots$. The binary vectors are treated as observed for this problem.

We present results in Figure 5.1 for 5,500 trials, where $\alpha_{\text{true}} \sim \text{Uniform}(1, 10)$ and $\gamma_{\text{true}} \sim \text{Uniform}(1, 10)$. We see that the inferred α_{out} and γ_{out} values center on the true α_{true} and γ_{true}, but increase in variance as these values increase. We believe that this is due in part to the reordering of the dimensions, which are not strictly decreasing in (5.5), though some reordering is necessary because of the nature of the two priors. We choose to generate data from (5.2) rather than (5.5) because it provides some added empirical evidence as to the correctness of the stick-breaking construction.

5.6.2 MNIST Handwritten Digits

We consider the digits 3, 5 and 8 using 1000 observations for each digit and projecting into 50 dimensions using PCA. We model the resulting digits matrix, $X \in \mathbb{R}^{50 \times 3000}$,

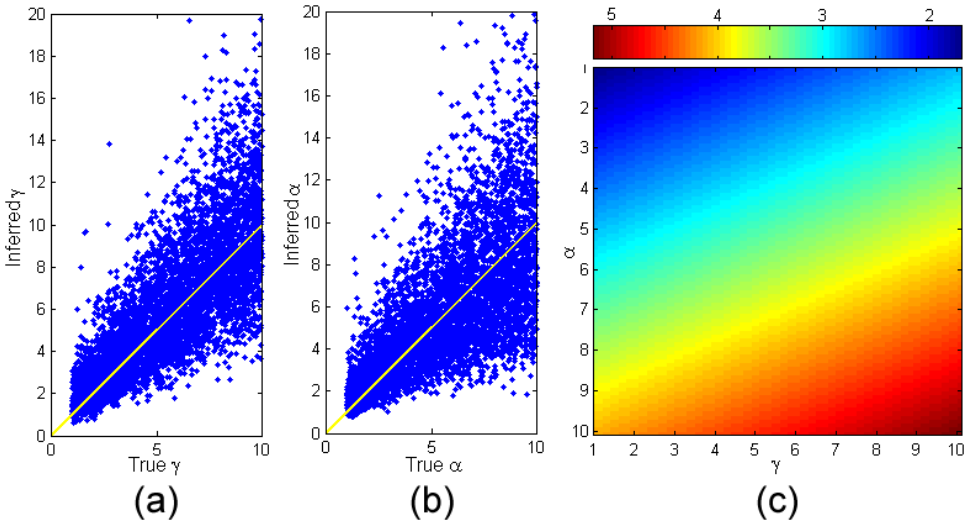


FIGURE 5.1: Synthetic results for learning α and γ. For each trial of 150 iterations, 10 samples were collected and averaged over the last 50 iterations. The step size $\Delta\alpha = 0.1$. (a) Inferred γ vs true γ (b) Inferred α vs true α (c) A plane, shown as an image, fit using least squares that shows the ℓ_1 distance of the inferred $(\alpha_{\text{out}}, \gamma_{\text{out}})$ to the true $(\alpha_{\text{true}}, \gamma_{\text{true}})$.

with a latent factor model [31, 52],

$$X = \Phi(W \circ Z) + E \tag{5.22}$$

where the columns of Z are samples from a Bernoulli process, and the elements of Φ and W are iid Gaussian. The symbol $\circ$ indicates element-wise multiplication. We infer all variance parameters using inverse-gamma priors, and integrate out the weights, w_n, when sampling z_n. Gibbs sampling is performed for all parameters, except for the variance parameters, where we perform variational inference [9]. We have found that the "inflation" of the variance parameters that results from the variational expectation leads to faster mixing for the latent factor model.

Figure 5.2 displays the inference results for an initialization of $K = 200$. The top-left figure shows the number of factors as a function of 10,000 Gibbs iterations, and the top-right figure shows the histogram of these values after 1000 burn-in iterations. For Monte Carlo integration, we use $S = 100{,}000$ samples from the stick-breaking

prior for sampling d_k and $p(z_{nk} = 1|\alpha, d_k, Z_{\text{prev}})$, and $S = 10{,}000$ samples for sampling α, since learning the parameter α requires significantly more overall samples. The average time per iteration was approximately 18 seconds, though this value increases when K increases and vice-versa. In the bottom two rows of Figure 5.2, we show four example factor loadings (columns of Φ), as well as the probability of its being used by a 3, 5 and 8.

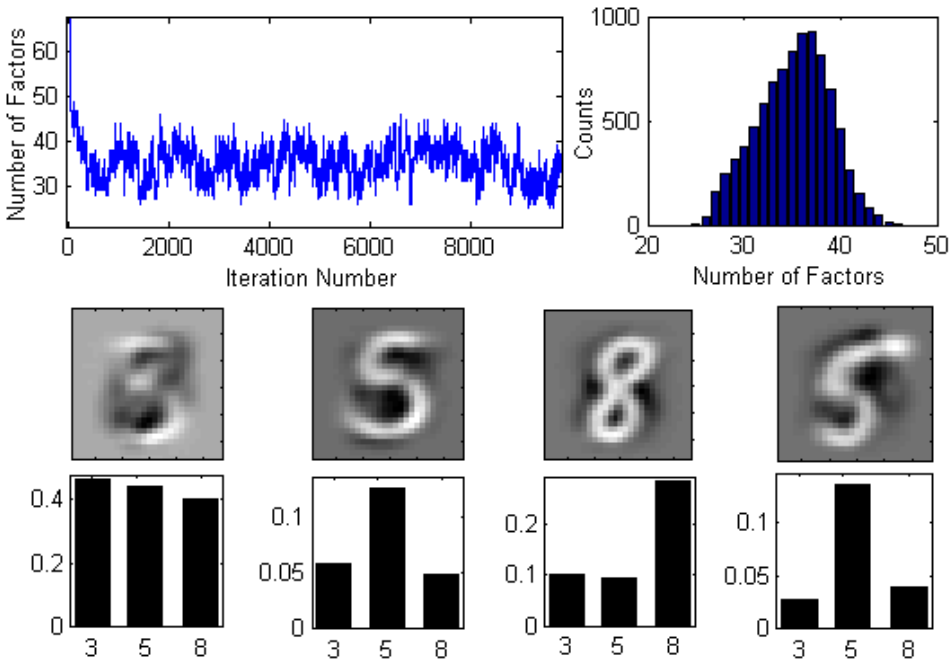


FIGURE 5.2: Results for MNIST digits 3, 5 and 8. Top left: The number of factors as a function of iteration number. Top right: A histogram of the number of factors after 1000 burn-in iterations. Middle row: Several example learned factors. Bottom row: The probability of a digit possessing the factor directly above.

5.6.3 Time-Evolving Gene Expression Data

We next apply the model discussed in Section 5.6.2 on data from a viral challenge study [81]. In this study, a cohort of 17 healthy volunteers were experimentally infected with the influenza A virus at varying dosages. Blood was taken at intervals between -4 and 120 hours from infection and gene expression values were extracted.

Of the 17 patients, 9 ultimately became symptomatic (i.e., became ill), and the goal of the study was to detect this in the gene expression values *prior* to the initial showing of symptoms. There were a total of 16 time points and 267 gene expression extractions, each including expression values for 12,023 genes. Therefore, the data matrix $X \in \mathbb{R}^{267 \times 12023}$.

In Figure 5.3, we show results for 4000 iterations; each iteration took an average of 2.18 minutes. The top row shows the number of factors as a function of iteration, with 100 initial factors, and histograms of the overall number factors, and the number of factors per observation. In the remaining rows, we show four discriminative factor loading vectors, with the statistics from the 267 values displayed as a function of time. We note that the expression values begin to increase for the symptomatic patients prior to the onset of symptoms around the 45th hour. We list the top genes for each factor, as determined by the magnitude of values in W for that factor. In addition, the top three genes in terms of the magnitude of the four-dimensional vector comprising these factors are RSAD2, IFI27 and IFI44L; the genes listed here have a significant overlap with those in the literature [81].

As motivated in [31], the values in Z are an alternative to hard clustering, and in this case are useful for group selection. For example, sparse linear classifiers for the model $y = X\beta + \epsilon$, such as the RVM [9], are prone to select single correlated genes from X for prediction, setting the others to zero. In [76], latent factor models were motivated as a dimensionality reduction step prior to learning the classifier $y = \Phi\hat{\beta} + \epsilon_2$, where the loading matrix replaces X and unlabeled data are inferred transductively. In this case, discriminative factors selected by the model represent groups of genes associated with that factor, as indicated by Z.

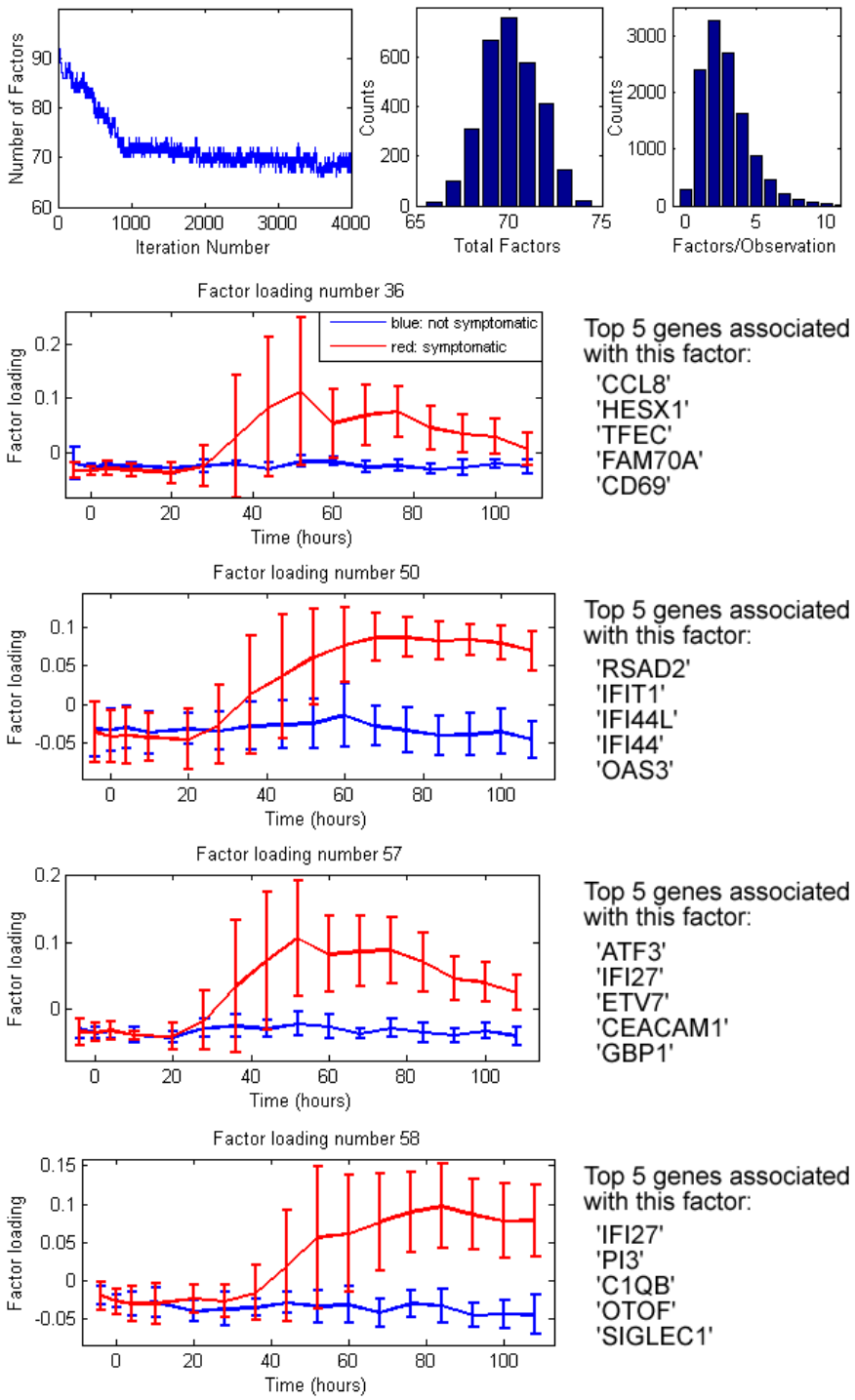


FIGURE 5.3: Results for time-evolving gene expression data. Top row: (left) Number of factors per iteration (middle) Histogram of the total number of factors after 1000 burn-in iterations (right) Histogram of the number of factors used per observation. Rows 2-5: Discriminative factors and the names of the most important genes associated with each factor (as determined by weight).

5.7 Conclusion

In this chapter, we have presented a stick-breaking construction of the beta process. The derivation of the construction relies heavily upon the constructive definition of the beta distribution, a lesser-known special case of [68], which has been exclusively used for the infinite Dirichlet process in the machine learning community. We presented an inference method that uses Monte Carlo integration, which aids inference by eliminating several random variables, and where the parameters α and γ are inferred. Results were presented on three data sets: (*i*) Synthetic data for learning α and γ; (*ii*) the MNIST handwritten digits 3, 5 and 8 using a latent-factor model; and (*iii*) time-evolving gene expression data for nonparametric learning of latent gene groupings, and for discovering discriminative factor loadings for classification.

As a final comment, we note that the limit of the representation in (5.2) reduces to the original IBP when $\alpha = 1$. Therefore, the stick-breaking process in (5.5) should be equal in distribution to the process in [71] for this parametrization. The proof of this equality is an interesting question for future work.

5.8 Appendix

Following $i-1$ breaks from a Beta$(1, \alpha)$ stick-breaking process, the remaining length of the unit-length stick is $\epsilon_i = \prod_{j=1}^{i-1}(1 - V_j)$. Let $S_j := -\ln(1 - V_j)$. Then, since it can be shown that $S_j \sim$ Exponential(α), and therefore $\sum_{j=1}^{i-1} S_j \sim$ Gamma$(i-1, \alpha)$, the value of ϵ_i can be calculated using only one random variable,

$$\begin{aligned} \epsilon_i &= \mathrm{e}^{-T_i} \\ T_i &\sim \text{Gamma}(i-1, \alpha) \end{aligned}$$

Therefore, to draw $V_i \prod_{j=1}^{i-1}(1 - V_j) = \epsilon_i V_i$, one can sample $V_i \sim$ Beta$(1, \alpha)$ and ϵ_i as above.

6

Image Interpolation Using Dirichlet and Beta Process Priors

6.1 Abstract

In this chapter, we present a Bayesian model for image interpolation and dictionary learning that uses two nonparametric priors for sparse signal representations: the beta process and the Dirichlet process. Additionally, the model uses spatial information within the image to encourage sharing of information within image subregions. We derive a hybrid MAP/collapsed Gibbs sampler, which performs collapsed Gibbs sampling for the latent indicator variables by integrating out several parameters and MAP estimation for all other parameters. We present experimental results, where we show an improvement over other state-of-the-art algorithms in the low-measurement regime.

6.2 Introduction

As we have discussed, Bayesian nonparametric analysis provides a powerful set of tools for modeling data, and has found extensive use in recent research (e.g., [73][82]

and references therein). A key advantage of these methods is the sparsity-promotion of the various nonparametric priors, which allows for many truncation issues to essentially be avoided. For example, the Dirichlet process [27] is useful for uncovering, or inferring the number of components in a mixture model, while the beta process [34] has recently found significant use for inferring the number of factors in latent factor models (see [54]).

In this chapter we present a Bayesian nonparametric algorithm for interpolating missing voxel values in incomplete images, including both natural and hyperspectral images [58]. The model uses the Dirichlet process and the beta process, as well as spatial information of pixel location within the image. Though we present the hierarchical structure for one model, we are actually presenting three models for the application considered in this chapter, the other two models being special cases of the presented model. We show results on complex, canonical images employed in the image processing community [45], and show an improvement in performance for high percentages of missing voxels, as well as the advantage provided by all three aspects of the proposed prior.

We present and discuss the model in Section 2 and inference equations in Section 3. We show experimental results in Section 4 and conclude in Section 5.

6.3 The Model

Let $\{y_n\}_{n=1}^N$ be a collection of N patches of size $m \times m \times p$ extracted from an image and reshaped into $P = pm^2$ dimensional vectors. Also, let $\{x_n\}_{n=1}^N$ be the two-dimensional coordinates for the corresponding patches, for example, the coordinates within the image of the upper-left pixel of the patch.

We model both y_n and x_n as being drawn from a mixture model having two modalities [53]. More precisely, we model each patch, y_n, as a sparse, weighted combination of a dictionary matrix, $\Phi \in \mathbb{R}^{P \times K}$, with additive noise, i.e., the BPFA

model of Chapter 4. Each patch location, x_n, is modeled as being generated from a mixture of Gaussians, G_d. The mixture model is therefore a joint mixture of BPFA models, and Gaussian mixture models. We observe that, removing the BPFA portion results in the nested Dirichlet process [65]. Using the notation H to represent the mixture model, we can write that $H = \sum_{k=1}^{\infty} \omega_k \delta_{\{\pi_d, G_d\}}$, where ω_k represent the mixing weights and the set $\{\pi_d, G_d\}$ represents the corresponding parameters.

The generative process of the complete (no missing) data set described above is,

$$\begin{aligned}
y_n &\sim \mathcal{N}\left(\Phi(w_n \circ z_n), \sigma_\epsilon^2 I\right) \\
x_n &\sim \text{GMM}\left(G_{c_n}\right) \\
w_n(k) &\sim z_n(k)\mathcal{N}\left(0, \sigma_w^2\right) + (1 - z_n(k))\delta_0 \\
z_n(k) &\sim \text{Bernoulli}\left(\pi_{c_n}(k)\right) \\
c_n &\sim \text{Multinomial}\left(\{1, \ldots, D\}, \eta\right) \\
\pi_d(k) &\sim \text{Beta}\left(\frac{\alpha\gamma}{K}, \alpha(1 - \frac{\gamma}{K})\right) \\
\phi_k &\sim \mathcal{N}(0, \sigma_\phi^2 I) \\
G_d &\sim \text{DP}(cG_0) \\
\eta &\sim \text{Dirichlet}\left(\frac{\beta}{D}, \ldots, \frac{\beta}{D}\right)
\end{aligned} \tag{6.1}$$

for $d = 1, \ldots, D$ and $k = 1, \ldots, K$. In the first line, the symbol $\circ$ indicates element-wise multiplication. Also, though the Dirichlet process and beta process are infinite-dimensional priors (i.e., K and D are infinite), we use finite-dimensional approximations, which work well in practice at finding sparse representations when K and D are large.

With reference to this model (6.1), the Dirichlet process is the prior on the D mixing weights, $\eta \in \Delta_D$, which are probabilities of using a particular component, (i.e., $\{\pi_d, G_d\}$ pair). Though each vector, π_d, is unique, they each correspond to the

same dictionary. The theoretical motivation that justifies a shared dictionary in the infinite limit is the hierarchical beta process [73], which has the same intuition as the hierarchical Dirichlet process. We do not build this into our prior, since simply imposing that the dictionary be shared and that the π_d be updated independently works as well as the HBP in practice.

The latent indicator, c_n, drawn from η determines the component from which patch y_n and pixel x_n come (i.e., determines that $\{\pi_{c_n}, G_{c_n}\}$ are used). The K-dimensional binary vector, z_n, generated using π_{c_n} then turns on or off dictionary elements for the n^{th} patch, and the weight vector, w_n, provides added flexibility. Drawing pixel locations from a Gaussian mixture model (written GMM$(\cdot)$ for short) imposes that patches that share a component must not only look alike via their usage of the dictionary, but also must be located in the same subregion of the image.

We finally note the "spike-slab" prior on $w_n(k)$. This is selected over $w_n \sim \mathcal{N}(0, \sigma_w^2 I)$ since it slightly increases the penalty for adding a dictionary element in the inference procedure, and allows for σ_w^2 to be updated using only the active elements of z_n, something we've found important in practice.

The two models that result as special cases are the BPFA model of chapter 4, which results from setting $D = 1$, and a mixture of BPFA models (with a shared dictionary), which results from eliminating the second modality. We consider these two models as well below.

6.3.1 Handling Missing Data

For interpolation and one step of inference, we integrate out the weight vector. We then use two properties of multivariate Gaussian distributions to handle the missing data. To review, if $y \sim \mathcal{N}(\Phi(w \circ z), \sigma_\epsilon^2 I)$ and $w \sim \mathcal{N}(0, \sigma_w^2 I)$, then integrating out w results in $y \sim \mathcal{N}(0, \sigma_\epsilon^2 I + \sigma_w^2 \Phi \text{diag}(z) \Phi^T)$, where diag$(z)$ forms a diagonal matrix using the vector z; results for the spike-slab prior employed above are the same.

Partitioning the vector y and the covariance matrix into their missing and observed parts, generically written as

$$\begin{bmatrix} y_{\mathrm{m}} \\ y_{\mathrm{o}} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \Sigma_{\mathrm{m}} & \Sigma_{\mathrm{m,o}} \\ \Sigma_{\mathrm{o,m}} & \Sigma_{\mathrm{o}} \end{bmatrix}\right) \tag{6.2}$$

then integrating out the missing data produces $y_{\mathrm{o}} \sim \mathcal{N}(0, \Sigma_{\mathrm{o}})$, which is the distribution used in likelihood calculations. For interpolation, the distribution of y_{m} given y_{o} is,

$$y_{\mathrm{m}}|y_{\mathrm{o}} \sim \mathcal{N}(\Sigma_{\mathrm{m,o}}\Sigma_{\mathrm{o}}^{-1}y_{\mathrm{o}}, \Sigma_{\mathrm{m}} - \Sigma_{\mathrm{m,o}}\Sigma_{\mathrm{o}}^{-1}\Sigma_{\mathrm{o,m}})$$

These two properties are applied to the relevant partitions of y_n and $\Sigma_n := \sigma_\epsilon^2 I + \sigma_w^2 \Phi \mathrm{diag}(z_n) \Phi^T$.

To fix notation in what follows, let Y be the $P \times N$ matrix formed by combining all y_n vectors. We define the set I_n^c containing the indices of measured values for the n^{th} column of Y and similarly define I_p^r for the p^{th} row of Y. For vectors, $v(I^c)$ selects dimensions of v, while for matrices, A_{I^c,I^r} selects rows and columns of A.

6.4 Model Inference

For model inference, we use both MAP estimation and Gibbs sampling;[1] we sample the latent component indicators, c_n, and latent binary indicators, z_n, and perform MAP updates for all other parameters. When sampling c_n and z_n, we integrate out, or collapse the values of the mixing weights, η, and the latent factor probabilities, π_d. The collapsing of the mixing weights follows the discussion in Section 1.3, and the latent factor probabilities follow the two-dimensional analogue of this discussion. A key difference in the sampling method below is that we do not update the collapsed probabilities with each sample to save computation time. Our approach is therefore

[1] Code is available at www.ee.duke.edu/∼jwp4/ICIP2010

equivalent to taking the mean of the posterior distributions of η and π_d as the values for these parameters for each iteration.

Full conjugacy within the hierarchical structure allows for Gibbs sampling to be performed throughout, as well as variational inference [9], as two alternative inference approaches. We also place conjugate inverse-gamma priors on all variance parameters and perform MAP inference for these values as well. These variance parameters are the Bayesian equivalent of regularization terms in optimization algorithms, and learning them significantly improves the performance of the model.

6.4.1 Maximum A Posteriori Updates and Collapsed Probabilities

MAP update for w_n:

$$w_n = \left(\frac{\sigma_\epsilon^2}{\sigma_w^2} I + \Phi_{I_n^c,:}^T \Phi_{I_n^c,:} \circ z_n z_n^T\right)^{-1} \text{diag}(z_n) \Phi_{I_n^c,:}^T y_n(I_n^c) \tag{6.3}$$

These updates are ℓ_2-regularized least squares solutions [9] calculated using only the activated dictionary elements for the current observation, as indicated by z_n.

MAP update for Φ: We define the matrix W in a similar manner as Y in Section 6.3.1. The p^{th} dimension of the updated dictionary is,

$$\Phi_{p,:} = Y_{p,I_p^r} W_{:,I_p^r}^T \left(\frac{\sigma_\epsilon^2}{\sigma_\phi^2} I + W_{:,I_p^r} W_{:,I_p^r}^T\right)^{-1} \tag{6.4}$$

We note that W will have zeros in the same locations as Z, and hence we do not have to write $W \circ Z$. The diagonal prior covariance in the dictionary allows for this analytical result.

MAP update for G_d: The mixing weights, means and covariances of G_d are calculated using all x_n for which the indicator $c_n = d$. We use a finite-dimensional

approximation to the DP; update equations can be found in [9].

Collapsing π_d: We integrate out the values of π_d to obtain the value denoted $\hat{\pi}_d$,

$$\hat{\pi}_d(k) = \frac{\frac{\alpha\gamma}{K} + \sum_{n=1}^{N} z_n(k)\mathbb{I}(c_n = d)}{\alpha + \sum_{n=1}^{N} \mathbb{I}(c_n = d)} \tag{6.5}$$

for $k = 1, \ldots, K$.

Collapsing η: We collapse, or integrate out the mixing weights to obtain $\hat{\eta}(d)$,

$$\hat{\eta}(d) = \frac{\frac{\beta}{D} + \sum_{n=1}^{N} \mathbb{I}(c_n = d)}{\beta + N} \tag{6.6}$$

for $d = 1, \ldots, D$.

6.4.2 Gibbs Sampling of Latent Indicators

Sample c_n: The latent component indicator is sampled from a D-dimensional multinomial distribution, with

$$p(c_n = d|\Theta) \propto p(z_n|\hat{\pi}_d)p(x_n|G_d)p(c_n = d|\hat{\eta}) \tag{6.7}$$

where $p(z_n|\hat{\pi}_d) = \prod_{k=1}^{K} \hat{\pi}_d(k)^{z_n(k)}(1 - \hat{\pi}_d(k))^{1-z_n(k)}$, $p(c_n = d|\eta) = \hat{\eta}(d)$ and $p(x_n|G_d)$ is the likelihood calculated using the GMM G_d. The symbol Θ represents the set of all parameters and latent indicators.

Sample z_n: For the sampling of the latent binary indicators, we integrate out the corresponding weights, w_n, in the way discussed in Section 6.3.1. Let $\tilde{z}_n$ be the binary indicator vector of the previous iteration. Using the definitions

$$M_n := \sigma_\epsilon^2 I + \sigma_w^2 \Phi_{I_n^c,:} \operatorname{diag}(\tilde{z}_n) \Phi_{I_n^c,:}^T \tag{6.8}$$

$$\xi_n^k := \sigma_w^2 \phi_k(I_n^c)^T M_n^{-1} \phi_k(I_n^c) \tag{6.9}$$

then,

$$\ln p(z_n(k) = 1|\Theta) \propto \ln \hat{\pi}_{c_n}(k) \; - \; \mathbb{I}\left(\tilde{z}_n(k) = 0\right) \times$$

$$\frac{1}{2}\left(\ln(1+\xi_n^k) - \frac{\sigma_w^2(\phi_k(I_n^c)^T M_n^{-1} y_n(I_n^c))^2}{1+\xi_n^k} + \ln 2\pi\sigma_w^2\right) \tag{6.10}$$

$$\ln p(z_n(k) = 0|\Theta) \propto \ln\left(1 - \hat{\pi}_{c_n}(k)\right) - \mathbb{I}\left(\tilde{z}_n(k) = 1\right) \times$$

$$\frac{1}{2}\left(\ln(1-\xi_n^k) + \frac{\sigma_w^2(\phi_k(I_n^c)^T M_n^{-1} y_n(I_n^c))^2}{1-\xi_n^k} - \ln 2\pi\sigma_w^2\right) \tag{6.11}$$

One of the two indicators will be active, which accounts for the effect of either the addition or subtraction of both a dictionary element and a $w_n(k)$ term ($\pi = 3.141...$ in $\ln 2\pi\sigma_w^2$). We note that the matrix inversion lemma and a property of matrix determinants were used in this derivation.

We use an approximation in inference by not updating the vector $\tilde{z}_n$ element by element, but rather sampling all K dimensions at once using the binary vector of the previous iteration. This saves significant computation time, since the matrix M_n does not need to be updated for each element in z_n that switches a value. To avoid this approximation while still seeking to maximize computation speed, rank one updates of the inverse of M_n can be performed, which we give below for completeness. Consider the matrix M_n in (6.8). If a value in $\tilde{z}_n$, say $\tilde{z}_n(j)$ is switched from 1 to 0, then this matrix is updated to

$$M_n^- = M_n - \sigma_w^2 \phi_j^{(n)} \phi_j^{(n)T}$$

where $\phi_j^{(n)}$ is the j^{th} column of $\Phi_{I_n^c,:}$. If $\tilde{z}_n(j)$ is switched from 0 to 1, this matrix is updated to

$$M_n^+ = M_n + \sigma_w^2 \phi_j^{(n)} \phi_j^{(n)T}$$

The inverses of these respective matrices are

$$(M_n^-)^{-1} = M_n^{-1} + \frac{M_n^{-1}\phi_j^{(n)}\phi_j^{(n)T}M_n^{-1}}{\sigma_w^{-2} - \phi_j^{(n)T}M_n^{-1}\phi_j^{(n)}} \tag{6.12}$$

$$(M_n^+)^{-1} = M_n^{-1} - \frac{M_n^{-1}\phi_j^{(n)}\phi_j^{(n)T}M_n^{-1}}{\sigma_w^{-2} + \phi_j^{(n)T}M_n^{-1}\phi_j^{(n)}} \tag{6.13}$$

6.5 Related Algorithms

In this section, we review three algorithms with which we compare, the K-SVD [1], MOD [25] and iMMSE [51] algorithms, and discuss their relationships with the proposed model. Because the first two algorithms use the OMP algorithm [59], we review this algorithm as well.

6.5.1 Orthogonal Matching Pursuits

The OMP algorithm is a method for learning the coefficient vector, w, of a linear model $y = \Phi w + \epsilon$. This is done by sequentially selecting elements to "turn on" in w, allowing them to be non-zero, followed by a least squares update of all active values [33]. Let $\hat{w}^{(t)}$ be the t-dimensional sub-vector of w after step t. Also, let $\hat{\Phi}$ be the set of column vectors in Φ, $\hat{\Phi} = \{\phi_i\}$. Let $\Psi^{(t)}$ be the matrix containing the t vectors selected from $\hat{\Phi}$ after step t. Then the two steps of the OMP algorithm are,

$$\psi_{t+1} = \arg\max_{\phi_i \in \hat{\Phi}} \frac{|\phi_i^T(y - \Psi^{(t)}\hat{w}^{(t)})|}{\|\phi_i\|_2} \tag{6.14}$$

$$\hat{w}^{(t+1)} = \left(\Psi^{(t+1)T}\Psi^{(t+1)}\right)^{-1}\Psi^{(t+1)T}y \tag{6.15}$$

The first line selects the vector from $\hat{\Phi}$ that is most correlated with the error of the current approximation. The second line then updates the coefficient vector by finding the least squares approximation using the updated matrix, $\Psi^{(t+1)}$. Because the error is at all times orthogonal to the subspace spanned by this matrix, unique

vectors are guaranteed to be selected from $\hat{\Phi}$ at each step. Upon termination at step T, the values of $\hat{w}^{(T)}$ are placed in their respective locations in the vector w.

6.5.2 Method of Optimal Directions

The MOD algorithm is a method for factorizing matrices $Y \in \mathbb{R}^{M \times N}$ into a matrix product plus noise, $Y = \Phi W + E$, where $\Phi \in \mathbb{R}^{M \times K}$. The value of K is to be set an can be larger than M, in which case an overcomplete dictionary is learned. This overcomplete dictionary is possible because the matrix W is defined to be sparse, with many values exactly equal to zero. The OMP algorithm is used to achieve this. The MOD algorithm requires a sparsity setting, T, which is the number of nonzero values for any column of W learned by the OMP algorithm. It then iterates between learning W with OMP for a fixed Φ, and finding the least squares solution to Φ for a fixed W. Let the function $w_n \leftarrow \text{OMP}(y_n, \Phi, T)$ indicate that w_n is learned using OMP with a preset sparsity level of T (i.e., the OMP algorithm of the previous section terminates when $t = T$). Then the MOD algorithm iterates between the following two steps, with each iteration indexed by i.

$$w_n^{(i+1)} \leftarrow \text{OMP}(y_n, \Phi^{(i)}, T), \quad n = 1, \ldots, N \tag{6.16}$$

$$\Phi^{(i+1)} = Y W^{(i+1)T} \left(W^{(i+1)} W^{(i+1)T} \right)^{-1} \tag{6.17}$$

The value of T should be set to $T < M$, and preferably $T \ll M$ to find a sparse representation. The precise value of this parameter can be found using cross-validation [33], which can be time consuming. To handle missing data, only the dimensions containing data are used for the OMP algorithm, and the update of Φ follows equation (6.4), with the exception that the regularization term in the inverse is removed.

6.5.3 K-SVD

The K-SVD algorithm is a simple modification of the MOD algorithm, the difference being that OMP is used to find the sparsity pattern, but once this is found, least

squares values for Φ and W are found simultaneously, with the elements in W being allowed to take nonzero values determined by the OMP algorithm. This algorithm can be broken into two section, the first being the OMP algorithm.

$$w_n^{(i+1)} \leftarrow \text{OMP}(y_n, \Phi^{(i)}, T), \quad n = 1, \ldots, N \tag{6.18}$$

$$z_n(k)^{(i+1)} = \mathbb{I}(w_n(k) \neq 0), \quad k = 1, \ldots, K \tag{6.19}$$

The binary matrix Z is the same size as W and an element in Z indicates whether the corresponding element in W is nonzero. Given Z, the second step iterates between updating Φ and W with the objective of minimizing the function

$$\Phi^{(i+1)}, W^{(i+1)} = \arg\min_{\Phi, W} \|Y - \Phi(W \circ Z)\|_{\text{F}} \tag{6.20}$$

where F indicates the Frobenious, or matrix ℓ_2 norm. For this second step, we suppress the iteration index, i, for clarity, and index the following iterations by j. Note that these sub-iterations take place within the i^{th} overall iteration.

$$w_n^{(j+1)} = \left(\Phi^{(j)T}\Phi^{(j)} \circ z_n z_n^T + \text{diag}(1 - z_n)\right)^{-1} \text{diag}(z_n)\Phi^{(j)T} y_n \tag{6.21}$$

$$\Phi^{(j+1)} = YW^{(j+1)T}\left(W^{(j+1)}W^{(j+1)T}\right)^{-1} \tag{6.22}$$

As with the MOD algorithm, to handle missing data, only the measured dimensions can be used for the OMP algorithm and equation (6.21) and equation (6.22) follows the un-regularized version of equation (6.4). The resulting Φ and W are not unique; for example, multiplying the first column of Φ and dividing the first row of W by the same number does not change the objective function in (6.20). However, this scaling does not change the *direction* of the first column of Φ, which is the only information used by OMP for learning a sparsity pattern, and therefore does not effect the algorithm. We note that the addition of $\text{diag}(1 - z_n)$ to equation (6.21) allows for the inverse to be defined, while not changing the desired result. We also

observe that equation (6.21) is an un-regularized version of equation (6.3) in the proposed algorithm.

Therefore, the proposed algorithm and K-SVD are very similar. The updates for Φ and W follow essentially the same steps, with the proposed algorithm including an ℓ_2 regularization term. Both algorithms also include a step where the sparsity pattern is learned and fixed. However, while K-SVD achieves this using the OMP algorithm, the proposed method uses a Bayesian interpretation of the problem and samples binary values from an appropriate Bernoulli distribution. This is an advantage of the proposed method, since the columns of Y can be constructed using sparse combinations of the columns of Φ with varying levels of sparsity. Also, the Bayesian framework allows for additional priors on all variance (i.e., regularization) parameters, which allows for an adaptive learning of the ℓ_2 penalty terms. This Bayesian framework also allows for more complicated model structures, such as the mixture model employed here with a second modality accounting for spatial information.

6.5.4 Iterative Minimum Mean Squared Error

The iMMSE algorithm is the final method considered and does not use the OMP algorithm. The iMMSE algorithm works by iteratively increasing the rank of the factorization of Y, followed by a minimizing of the squared error to all measured values. For a rank r factorization, this is written as

$$Y = \Phi^{(r)} W^{(r)} + E$$

where W here is not constrained to be sparse. Given a fixed rank, r, the updates for $\Phi^{(r)}$ and $W^{(r)}$ iterate between

$$W^{(r)} = \left(\Phi^{(r)T}\Phi^{(r)}\right)^{-1}\Phi^{(r)T}Y \tag{6.23}$$

$$\Phi^{(r)} = YW^{(r)T}\left(W^{(r)}W^{(r)T}\right)^{-1} \tag{6.24}$$

For missing data scenarios, as we have here, only the observed dimensions for any y_n, and the corresponding dimensions of $\Phi^{(r)}$ are used. The final rank is found by terminating the algorithm when the squared error falls below a preset threshold.

6.6 Experiments

RGB Image Interpolation

We first show experimental results for the image in Figure 6.3, also used in [45]. We extracted $5 \times 5 \times 3$ overlapping patches from the image centered on each pixel for which the patch does not fall outside the image. No prior training is performed on separate images to aid inference, as is done in [45], but rather all learning is done *in situ*. We also considered other patch sizes, for example $7 \times 7 \times 3$ (reconstruction PSNR = 29.25, compared with PSNR = 29.65 in [45], where prior training was done), and $8 \times 8 \times 3$ (PSNR = 29.47, compared with PSNR = 29.31 in [82], where Gibbs sampling was performed throughout), these PSNR values being for 80% of the voxels missing at random. As our intention is to compare performance between algorithms, we present results for $5 \times 5 \times 3$, noting that similar results were observed for other patch sizes.

We compare with five other algorithms in Figures 6.1 and 6.2: 1) The proposed model *without* spatial information, 2) The model in [82], which is the proposed model without the Dirichlet process or spatial information, 3) the K-SVD algorithm [1] *without* using a prior database, 4) the MOD algorithm [25] and 5) an iterative minimum MSE (iMMSE) algorithm [51] in which the dictionary size increases one element at a time, followed by a minimization of the squared error to all measured values.

As discussed, the K-SVD and MOD algorithms make extensive use of the OMP algorithm [59], which requires a sparsity setting, T, determining the number of dictionary elements to be used for each patch. Defining p to be the probability of a

missing pixel, we set this value to $T \approx .75 * (1-p)3m^2$, which is 3/4 of the expected number of measured voxels in a patch. For 80% missing, this value is $T = 11$. We ran 200 iterations of each algorithm (except iMMSE), which was sufficient to converge to a stable PSNR. For iMMSE, the stopping point (rank of the factorized matrix) can be determined by setting a threshold on the approximation error, or by viewing the image; we simply increase the dictionary and use the best PSNR result, which is not practical, but does give an empirical upper bound on performance.

In Figures 6.3 and 6.4, we present example reconstruction results, where the clustering clearly shows the spatial aspect of the prior; see [45] for original images. This helps explain the advantage of the model in the low-measurement regime, shown in Figure 6.1 for the castle image; patches with few measurements may have difficulty clustering based only upon their dictionary usage, and spatial information can improve this clustering by encouraging patches to cluster by region as well as appearance. Separating patches by region makes the usage of the dictionary less ambiguous by allowing patches to more effectively share statistical strength, which in turn aids in constructing a better dictionary. In Table 6.1, we show the cost of this added complexity in runtime. Similar performance was observed for other images, which we omit for space. We also show in Figure 6.5 the reconstructions for all algorithms considered for the 80% missing problem and similar results for the mushroom image in Figure 6.6. As can be seen, many algorithms perform well, with the benefit of the proposed model coming from the smoothness of the interpolation. Also, we note that in Figures 6.1 and 6.2, the PSNR is calculated using only the missing voxels.

In Figure 6.4, we show example reconstructions for 75% missing for the mushroom image. These images show the tendency of the proposed algorithm to produce smoother images. This explains why the reconstructions for the castle image was significantly better for the proposed algorithm (where the sky and water is smooth), while for the mushroom image, the results are more competitive. Several reconstruc-

tions for the castle image are shown in Figure 6.5. We reiterate that, for the iMMSE algorith, the performance is very dependent upon the rank, and we choose the best result for this algorithm, which requires information that a user would likely not have. On the other hand, all other algorithms considered converge to a result that does not change significantly between iterations. In Figure 6.8 we show the clustering results using *no* spatial information for the castle and mushroom images with 80% missing.

Hyperspectral Image Interpolation

We also show results for the proposed algorithm on a hyperspectral data cube of size $150 \times 150 \times 210$. We ran tests for $80\%, 85\%, 90\%$ and 95% missing and both $3 \times 3 \times 210$ and $4 \times 4 \times 210$ patch sizes. In general results for K-SVD and MOD were better *except* for the 95% missing case. Looking at the per-iteration run times for both patch sizes in Table 6.2 and Table 6.3, however, we see that the computation time for these two algorithms is likely to be prohibitively high for real-world applications. In the 95% missing scenario, the proposed algorithms outperformed K-SVD and MOD as we show in the following way.

In Figure 6.9 and Figure 6.10, we show the MSE over a stretch of 20 spectral bands that correspond to wavelengths with high return intensities. We see that the K-SVD and MOD algorithms perform poorly in this range (they are comparable in other ranges). We show examples of their corresponding reconstructed images in Figure 6.11 and Figure 6.12. As these figures show, the K-SVD and MOD tend to produce much noisier reconstructions. To reiterate, though this issue is not present for higher measurement scenarios, the computational time required for the K-SVD and MOD algorithms are prohibitively slow. Full disclosure: No algorithm outperformed the iMMSE algorithm for the hyperspectral data, for which the MSE was significantly smaller.

6.7 Conclusion

We have presented a Bayesian nonparametric model for image interpolation. The model uses the beta process for dictionary learning, the Dirichlet process for flexibility in dictionary usage and spatial information within the image, which encourages similar dictionary usage within subregions of the image. Experiments on two rgb images showed an advantage of our model compared with other algorithms in the low-measurement regime. We also showed an advantage of the proposed method over the K-SVD and MOD algorithms for hyperspectral images when the number of measurements is very small.

	Time per iteration (minutes)			
	60%	70%	80%	90%
BP & DP & Spatial	3.52	3.04	2.67	2.30
BP & DP, No Spatial	2.23	1.81	1.38	0.96
BP Only	2.02	1.63	1.26	1.15
K-SVD	2.50	1.64	1.11	0.69
MOD	2.58	1.73	1.17	0.67

Table 6.1: Average per-iteration run time for algorithms as function of percent missing data (castle image). Comparison is not meaningful for the iMMSE algorithm (which is significantly faster).

	Time per iteration (minutes)			
	80%	85%	90%	95%
BP & DP & Spatial	16.1	10.6	5.9	2.1
BP & DP, No Spatial	18.6	10.4	5.5	1.8
BP Only	14.0	11.2	5.7	1.7
K-SVD	143.8	68.9	23.3	7.4
MOD	131.6	51.3	14.8	3.4

Table 6.2: Average per-iteration run time for algorithms as function of percent missing data for the hyperspectral image problem using $3 \times 3 \times 210$ patches.

	Time per iteration (minutes)			
	80%	85%	90%	95%
BP & DP & Spatial	49.7	29.9	13.6	4.4
BP & DP, No Spatial	48.5	28.3	14.6	4.5
BP Only	48.3	27.5	13.3	4.8
K-SVD	683.8	311.0	108.0	24.9
MOD	622.4	301.4	86.1	11.6

Table 6.3: Average per-iteration run time for algorithms as function of percent missing data for the hyperspectral image problem using $4 \times 4 \times 210$ patches.

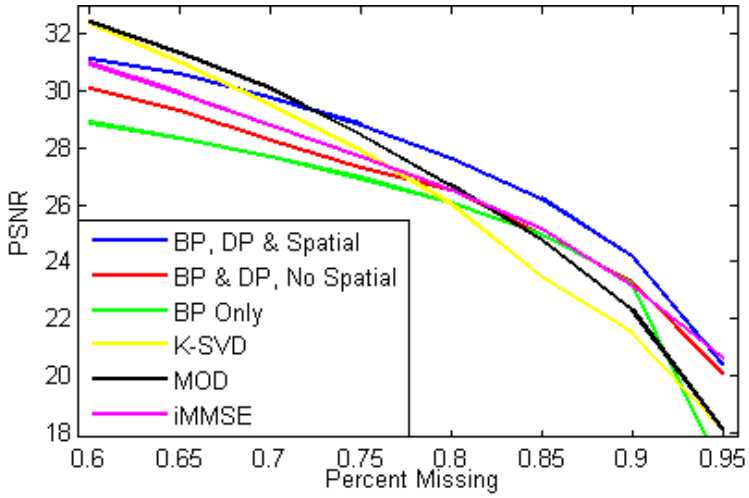


FIGURE 6.1: Castle image: PSNR of interpolated missing data using $5\times5\times3$ patches averaged over five trials. The proposed algorithm performs well for low-measurement percentages. We set $K = 100$ and $D = 50$.

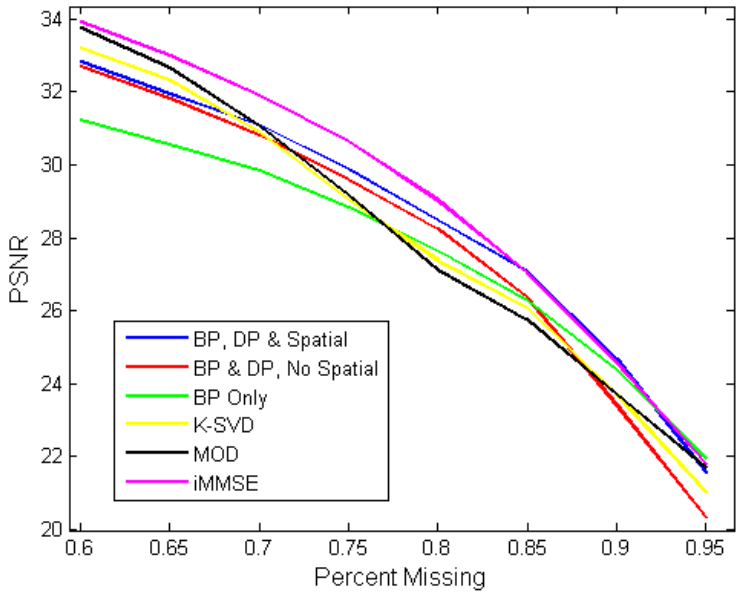


FIGURE 6.2: Mushroom image: PSNR of interpolated missing data using $5 \times 5 \times 3$ patches averaged over five trials. The proposed algorithm performs well for low-measurement percentages, but never better than iMMSE for this image. We set $K = 100$ and $D = 50$.

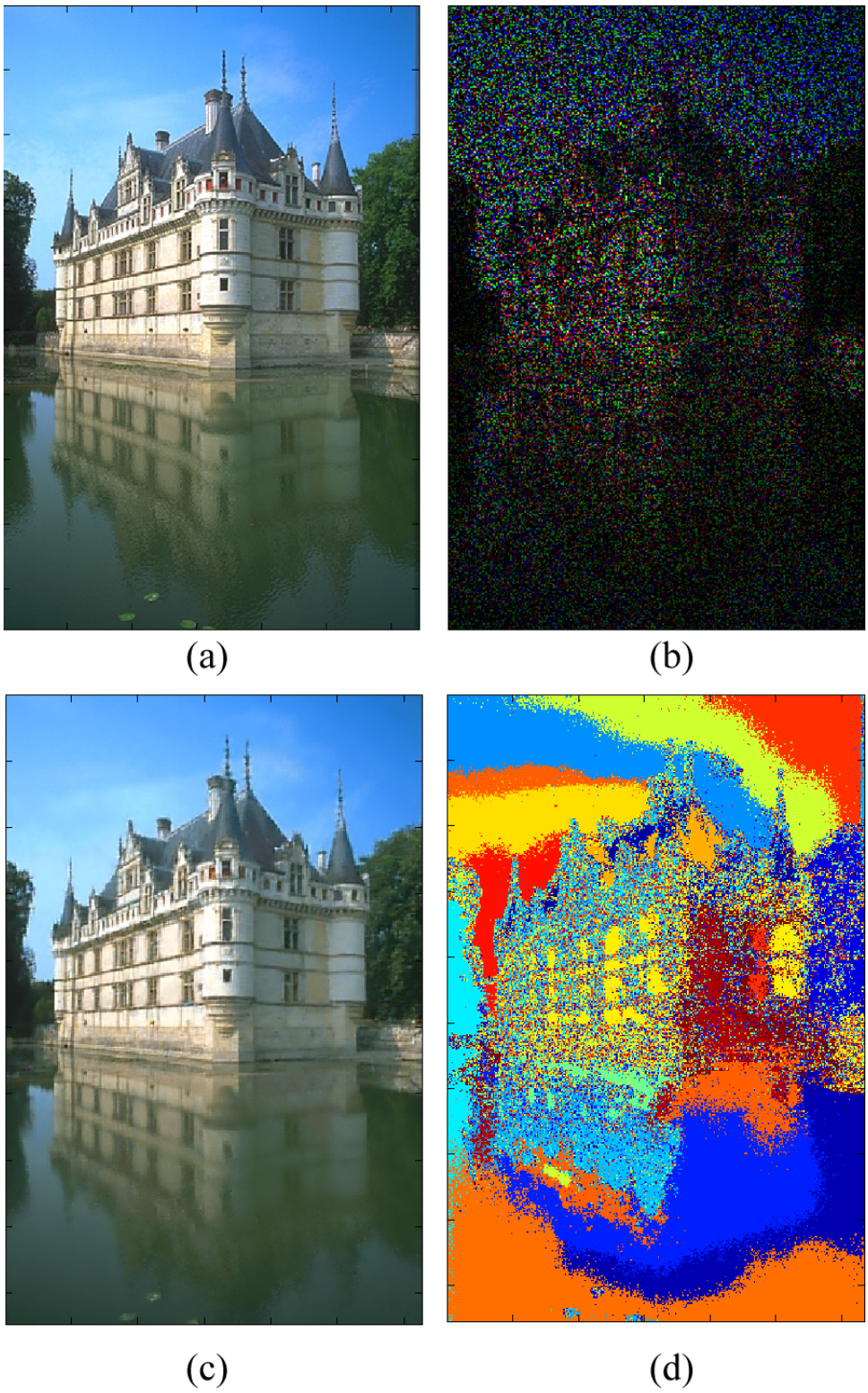


FIGURE 6.3: Example result (5×5×3 patch): (a) Original image, (b) 80% random missing, (c) Reconstructed image: PSNR = 28.76 (d) Clustering results: Cluster index as a function of pixel location.

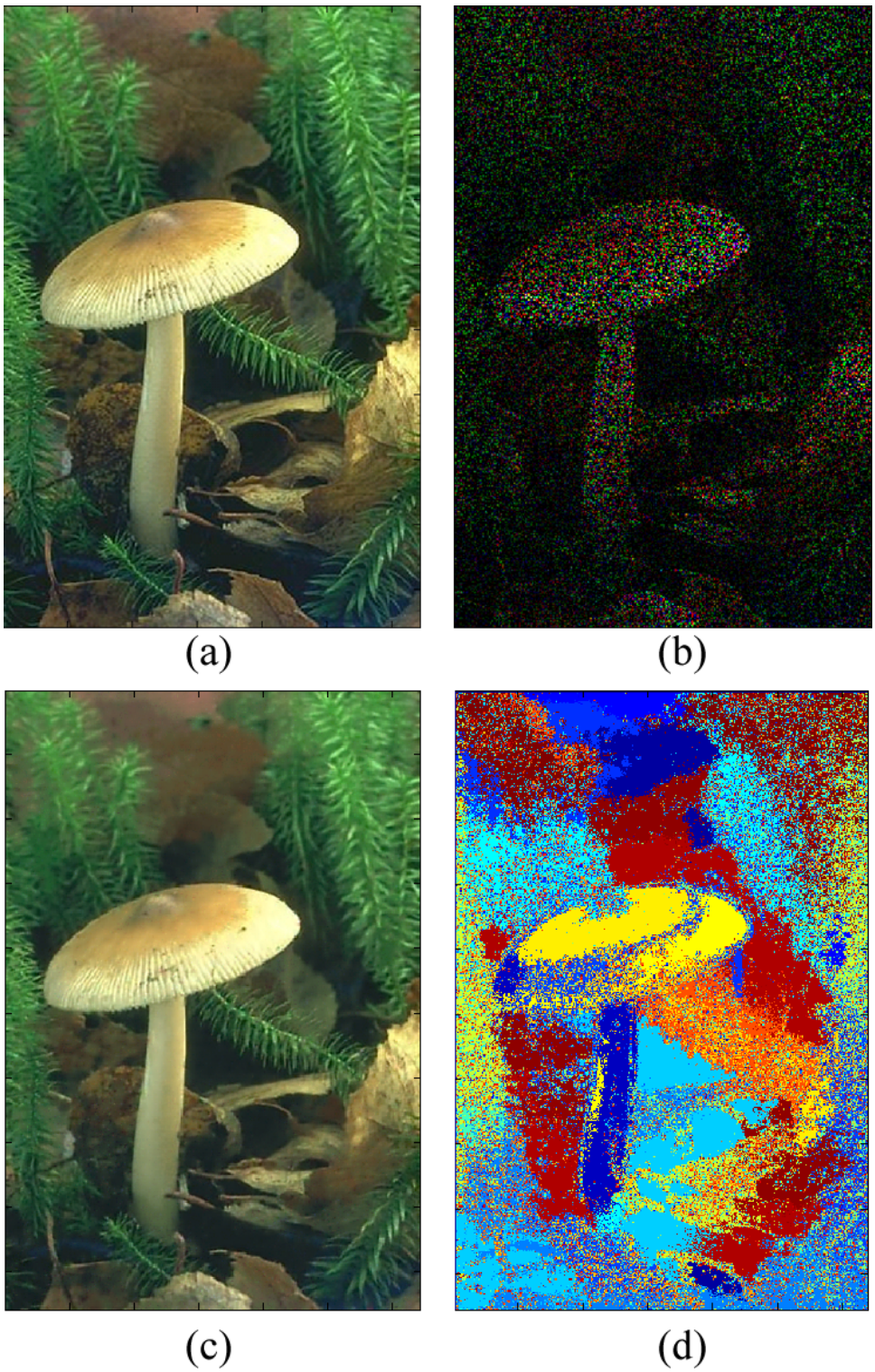


FIGURE 6.4: Example result (5×5×3 patch): (a) Original image, (b) 80% random missing, (c) Reconstructed image: PSNR = 28.76 (d) Clustering results: Cluster index as a function of pixel location.

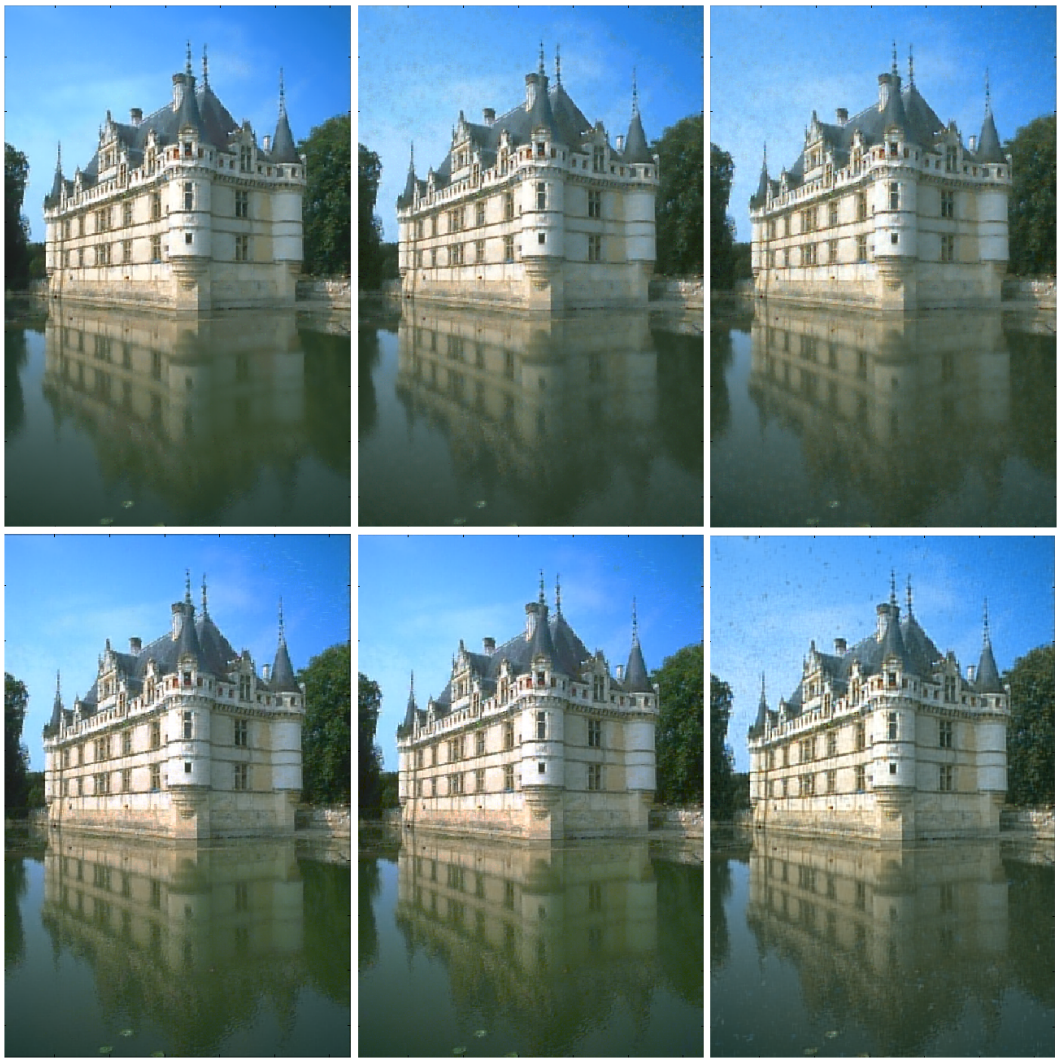


FIGURE 6.5: Reconstructions for 80% missing and $5 \times 5 \times 3$ patches for (top-left and clockwise) the proposed algorithm, the proposed algorithm without spatial information, the proposed algorithm without the DP, K-SVD, MOD and iMMSE.

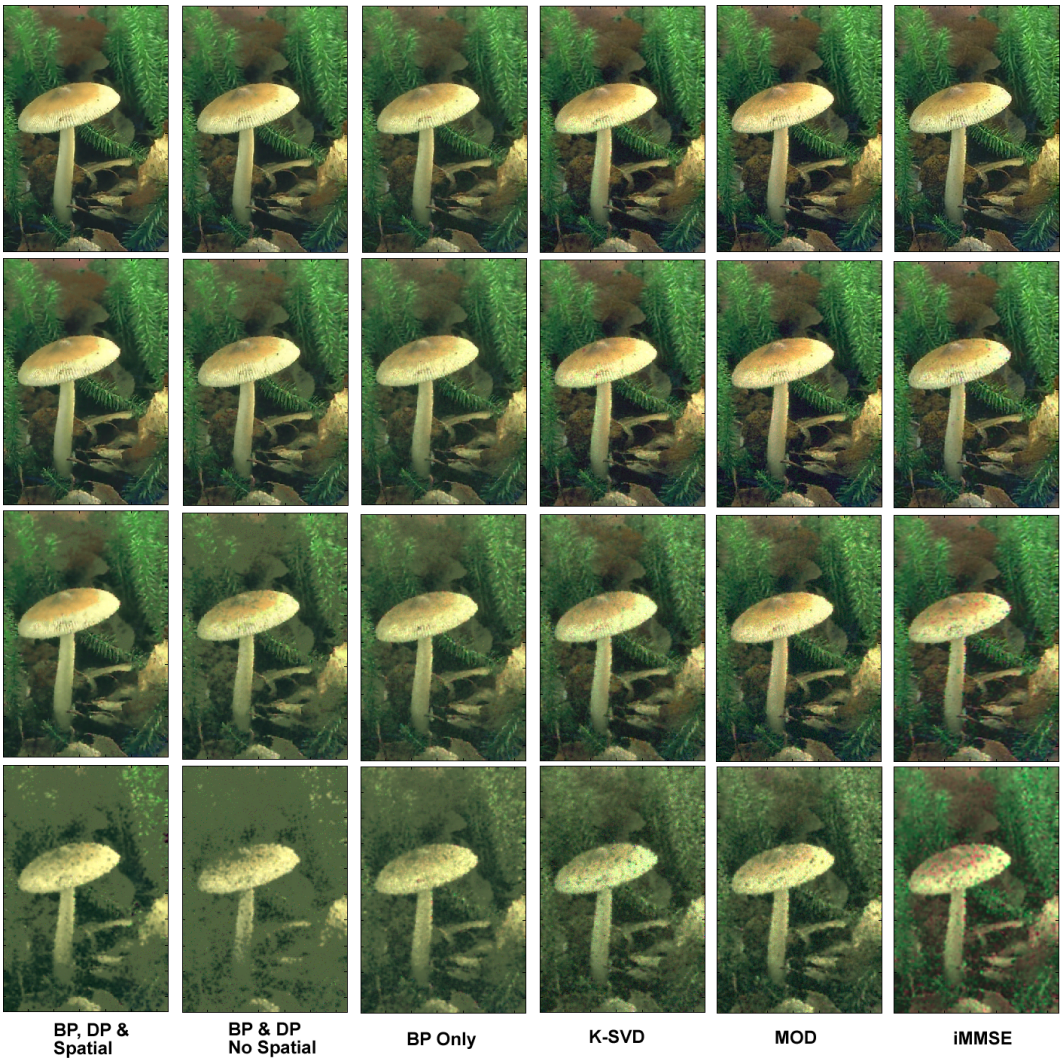


FIGURE 6.6: Reconstructions for (from top to bottom) 80%, 85%, 90% and 95% missing and $5 \times 5 \times 3$ patches for all algorithms considered.

FIGURE 6.7: Reconstruction results for 75% missing for (upper-right) iMMSE, (lower-left) Proposed algorithm, (lower-right) K-SVD.

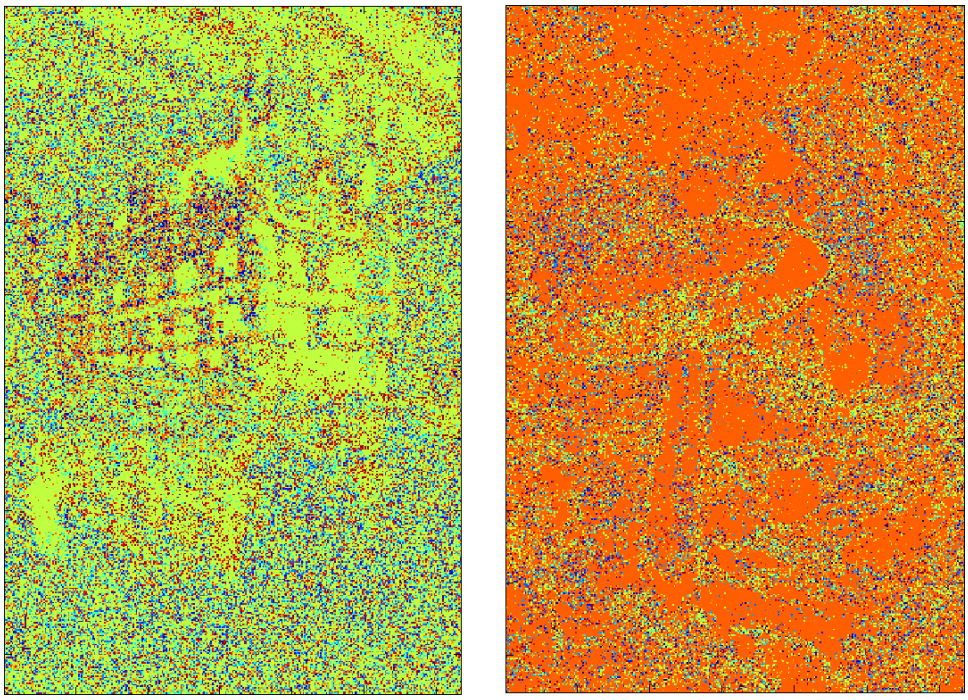


FIGURE 6.8: Clustering results (5×5×3 patch) using *no* spatial information for 80% missing: (left) Castle image, (right) Mushroom image. Because no spatial continuity is enforced in the DP prior, no spatially meaningful clustering takes place, which results in a slightly worse reconstruction.

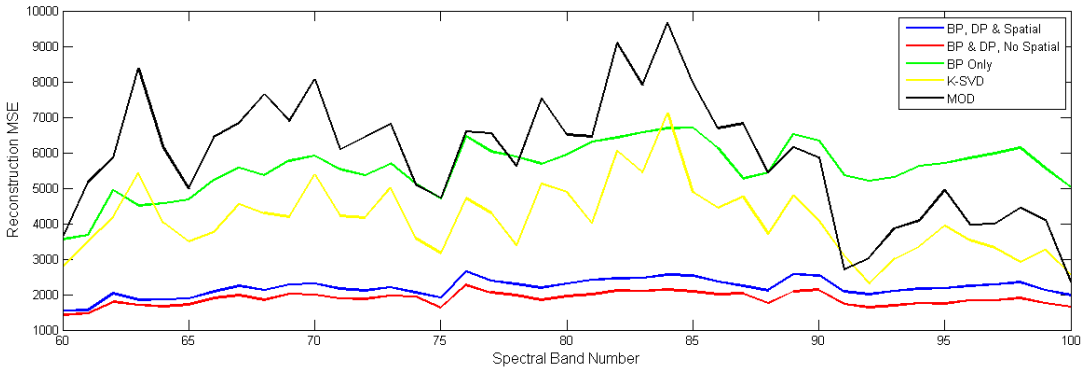


FIGURE 6.9: Hyperspectral Data: The MSE of the reconstruction using $3 \times 3 \times 210$ patches and 95% missing data. The plot shows the MSE over spectral band number 60 to 100.

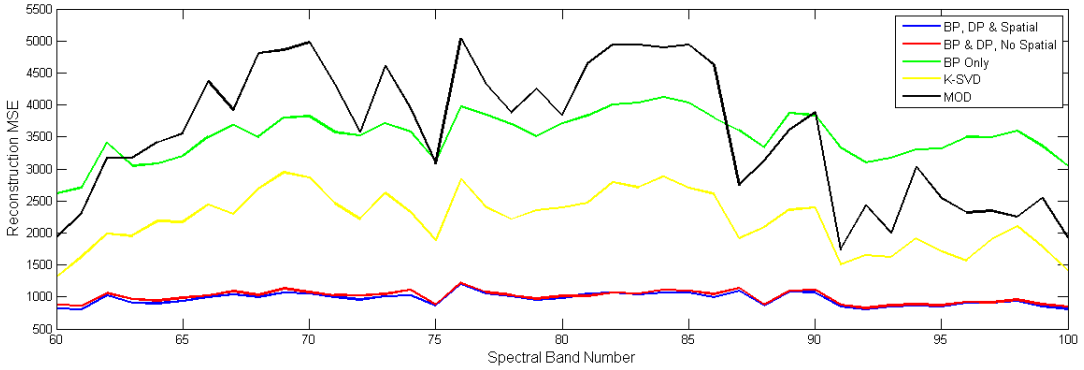


FIGURE 6.10: Hyperspectral Data: The MSE of the reconstruction using $4 \times 4 \times 210$ patches and 95% missing data. The plot shows the MSE over spectral band number 60 to 100.

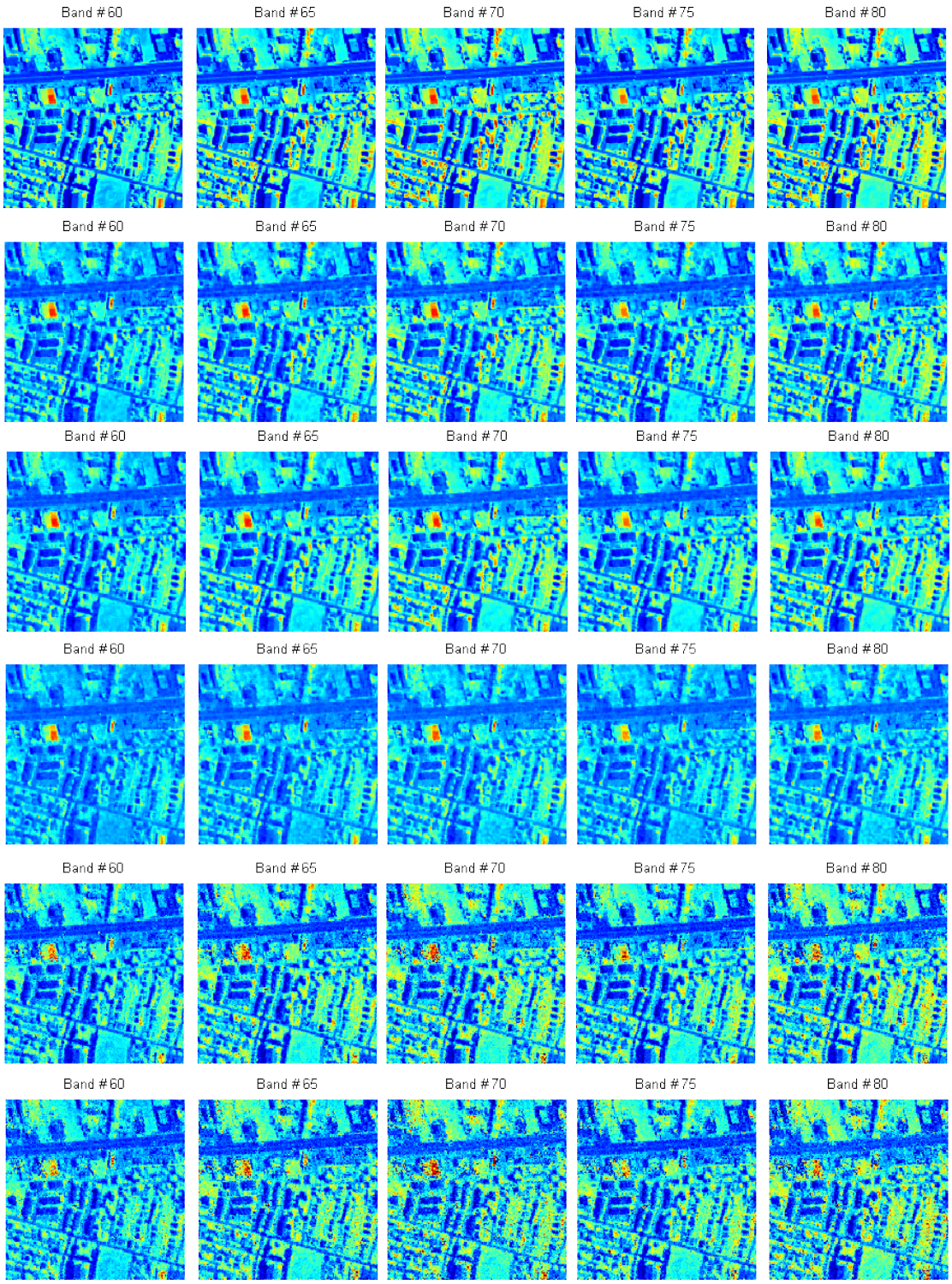


FIGURE 6.11: Reconstruction results for the indicated spectral bands using 3×3×210 patches. The plots according to row staring with the top are, 1. original data, 2. BP, DP & Spatial, 3. BP & DP, No Spatial, 4. BP Only, 5. K-SVD, 6. MOD.

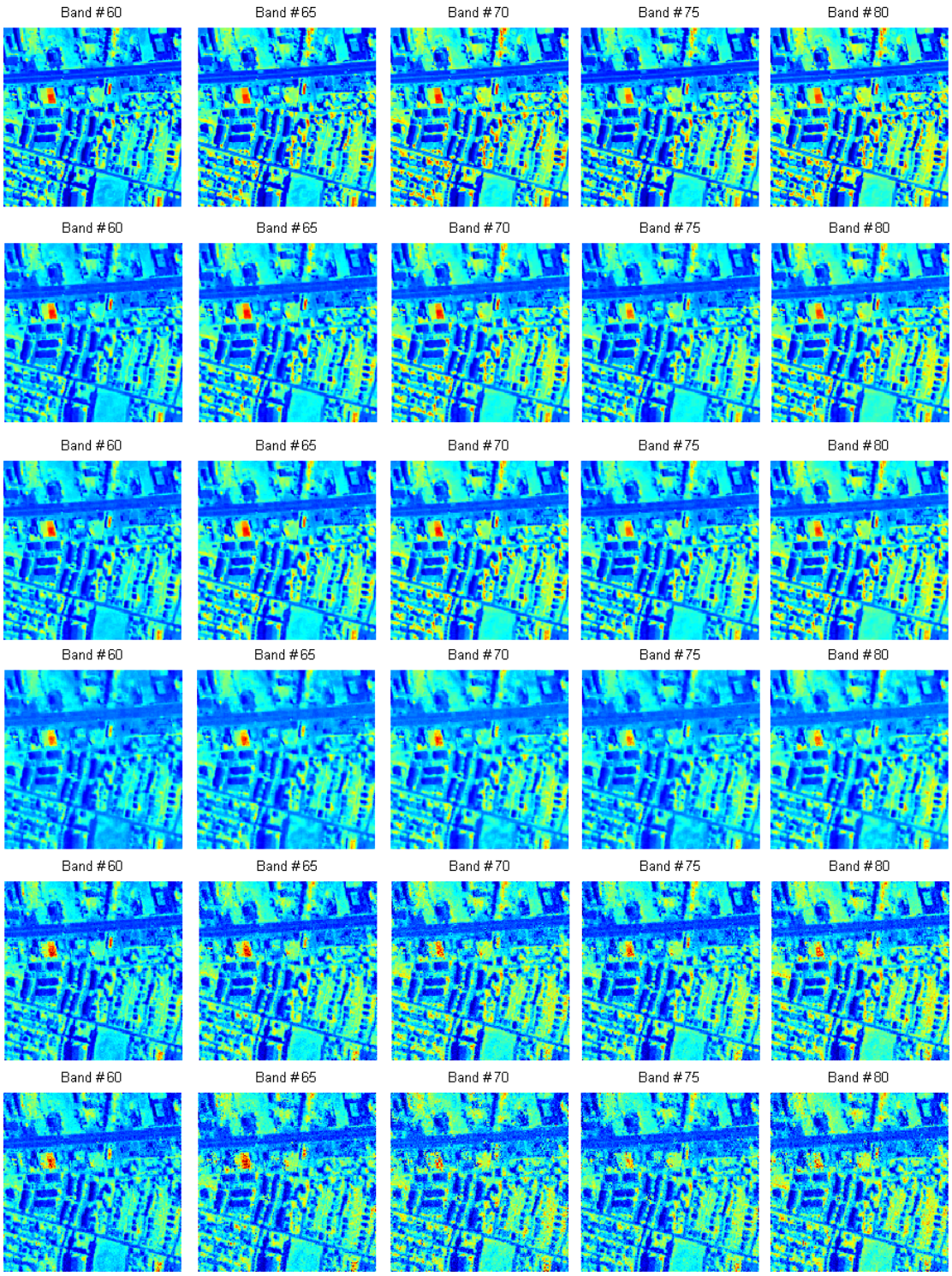


FIGURE 6.12: Reconstruction results for the indicated spectral bands using 4×4×210 patches. The plots according to row staring with the top are, 1. original data, 2. BP, DP & Spatial, 3. BP & DP, No Spatial, 4. BP Only, 5. K-SVD, 6. MOD.

7

Conclusion

This dissertation has focused on two Bayesian nonparametric priors, the Dirichlet and beta processes, for machine learning. The thesis was split evenly between theoretical results regarding these priors, and applications of these priors to problems such as compressive sensing, gene analysis and image interpolation. The focus was split evenly between the Dirichlet process and the beta process as follows:

In Chapter 1, we reviewed the Dirichlet process in detail and presented analytically derived values for the expectation and variance of the entropy of Dirichlet processes. In Chapter 2, we continued the discussion of Chapter 1 by looking more in-depth at Sethuraman's stick-breaking construction of a finite-dimensional Dirichlet prior [68]. In this chapter, we presented a novel comparison of the finite-symmetric Dirichlet distribution and the truncated Dirichlet process as mixture modeling priors, as well as two new applications of this construction for performing conjugate inference for the concentration parameter of a Dirichlet distribution, as well as the hierarchical Dirichlet process [70]. In Chapter 3, we extended the framework of Dirichlet process priors to include data that has multiple modalities.

In Chapter 4, we moved to the beta process for Bayesian nonparametric learning of latent factor models. We presented a new variational inference algorithm for learning these models and applied the model to several data sets, including an application to compressive sensing. For this application, we showed the advantage of dictionary learning methods for CS inversion as opposed to using an off-the-shelf basis. In Chapter 5, we presented a new stick-breaking construction of the beta process. We believe this is a major theoretical contribution to the theory of beta processes, as the stick-breaking construction of the Dirichlet process was to Dirichlet processes. We give a proof of the construction, as well as a method for performing inference for this prior. In Chapter 6, we presented new models for image interpolation using both Dirichlet and beta process priors. The central model uses two modalities, the framework for this being presented in Chapter 4.

Bibliography

[1] M. Aharon, M. Elad and A. Bruckstein (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans on Sig Proc*, 54(11): 4311-4322.

[2] N. Ahmed, T. Natarajan and K.R. Rao (1974). Discrete cosine transform. *IEEE Trans. on Computers*, pp. 90-93.

[3] D. Aldous (1985). Exchangeability and related topics. *École d'ete de probabilités de Saint-Flour XIII-1983* 1-198 Springer, Berlin.

[4] C. Andrieu, N. de Freitas, A. Doucet and M.I. Jordan (2003). An introduction to MCMC for machine learning. *Machine Learning*, vol. 50, pp. 5-43.

[5] C.E. Antoniak (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Annals of Statistics*, 2:1152-1174.

[6] R. Baraniuk, M. Davenport, R. DeVore and M. Wakin (2008). A simple proof of the restricted isometry property for random matrices. *Constructive Approximation* pp. 253-263.

[7] D. Basu (1955). On statistics independent of a complete sufficient statistic. *Sankhya: The Indian Journal of Statistics*, 15:377-380.

[8] M.J. Beal (2003). *Variational Algorithms for Approximate Bayesian Inference* PhD thesis, Gatsby Computational Neuroscience Unit, University College London.

[9] C.M. Bishop (2006). *Pattern Recognition and Machine Learning*, Springer, New York.

[10] P. Billingsley (1995). *Probability and Measure, 3rd edition.* Wiley Press, New York.

[11] D. Blackwell and J.B. MacQueen (1973). Ferguson distributions via Pólya urn schemes. *The Annals of Statistics,* 1:353-355.

[12] D. Blei and M.I. Jordan (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, vol.3, pp. 993-1022.

[13] D. Blei and M.I. Jordan (2006). Variational inference for Dirichlet process mixtures. *Journal of Bayesian Analysis,* 1(1):121-144.

[14] D.M. Blei and P.I. Frazier (2009). Distance dependent Chinese restaurant processes. *Technical Report*.

[15] E.J. Candès and B. Recht (2008). Exact matrix completion via convex optimization, to appear in *Foundations of Computational Mathematics*.

[16] E.J. Candès and M. Wakin (2008). An introduction to compressive sampling. *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 2130.

[17] E.J. Candès and Y. Plan (2009). Matrix completion with noise. submitted to *Proc. of the IEEE*.

[18] R. Caruana (1997). Multitask learning. *Machine Learning*, 28:41-75.

[19] R.J. Connor and J.E. Mosimann (1969). Concepts of independence for proportions with a generalization of the Dirichlet distribution. *Journal of the American Statistical Association,* 64:194-206.

[20] S. Dasgupta and A. Gupta (1999). An elementary proof of the Johnson-Lindenstrauss lemma. Technical Report 99-006, UC Berkeley.

[21] A. Dempster, N. Laird and D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *Journal of Royal Statistical Society B*, vol. 39, no. 1, pp. 1-38, 1977.

[22] D.L. Donoho (2006). Compressed sensing, *IEEE Trans. on Information Theory*, 52 (4) : 1289-1306.

[23] D.B. Dunson and J.H. Park (2008). Kernel stick-breaking processes. *Biometrika*, vol. 95, no. 2, pp. 307-323.

[24] B. Efron, T. Hastie, I. Johnstone and R. Tibshirani (2004). Least angle regression (with discussion), *Annals of Statistics* vol. 32, no. 2, pp. 407-499.

[25] K. Engan, S.O. Aase and J.H. Huszy (1999). Method of optimal directions for frame design. *Proc. of ICASSP*, 5:2443-2446.

[26] M.D. Escobar and M. West (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association.* 90(430):577-588.

[27] T. Ferguson (1973). A Bayesian analysis of some nonparametric problems. *Annals of Statistics,* 1:209-230.

[28] B. de Finetti (1937). La prévision: ses lois logiques, ses sources subjectives. *Ann. Inst. H. Poincaré,* 7:1-68.

[29] D. Gamerman and H.F. Lopes (2006). *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference, Second Edition*, Chapman & Hall.

[30] Z. Ghahramani and T.L. Griffiths and P. Sollich (2007). Bayesian nonparametric latent feature models. *Bayesian Statistics.*

[31] T.L. Griffiths and Z. Ghahramani (2005). Infinite latent feature models and the Indian buffet process. *NIPS*, pp. 475-482.

[32] P. Halmos (1944). Random Alms. *The Annals of Mathematical Statistics,* 15:182-189.

[33] T. Hastie, R. Tibshirani and J. Friedman (2009). *The Elements of Statistical Learning, Second Edition.* Springer, New York.

[34] N.L. Hjort (1990). Nonparametric Bayes estimators based on beta processes in models for life history data. *Annals of Statistics*, 18(3):1259-1294.

[35] A.E. Hoerl and R.W. Kennard (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics, Special Ed., Feb. 2000*, 42(1):80-86.

[36] H. Ishwaran and L.F. James (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association,* 96:161-173.

[37] H. Ishwaran and M. Zarepour (2002). Dirichlet prior sieves in finite normal mixtures. *Statistica Sinica 12* pp. 941-963.

[38] H. Ishwaran and M. Zarepour (2002). Exact and approximate sum-representations for the Dirichlet process. *Canadian Journal of Statistics,* 30, 269-283.

[39] S. Ji, Y. Xue and L. Carin (2008). Bayesian compressive sensing. *IEEE Trans. on Signal Processing*, vol. 56.

[40] C.R. Johnson (1990). Matrix completion problems: a survey. *Proceed. of Symposia in Applied Mathematics*, 40:171-198.

[41] N. Johnson and S. Kotz (1977). *Urn Models and Their Applications*. Wiley Series in Probability and Mathematical Statistics.

[42] I. Jolliffe (2005). *Principal Component Analysis*, Wiley & Sons, New York.

[43] J.F.C. Kingman (1993). *Poisson Processes*. Oxford Studies in Probability 3.

[44] D. Knowles and Z. Ghahramani (2007). Infinite sparse factor analysis and infinite independent components analysis. *7th International Conference on Independent Component Analysis and Signal Separation*.

[45] J. Mairal, M. Elad and G. Sapiro (2008). Sparse representation for color image restoration. *IEEE Trans. Image Proc.*, vol. 17.

[46] S. Mallat (1999). *A Wavelet Tour of Signal Processing*. Eselvier, London, UK.

[47] E. Meeds, Z. Ghahramani, R. Neal and S. Roweis (2007). Modeling dyadic data with binary latent factors. *NIPS*, pp. 977-984.

[48] P. Muliere and L. Tardella (1998). Approximating distributions of random functionals of Ferguson-Dirichlet priors. *Canadian Journal of Statistics*, 26,283-297.

[49] R.M. Neal (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics,* 9(2):249-265.

[50] K. Ni, J. Paisley, L. Carin and D. Dunson (2008). Multi-task learning for analyzing and sorting large databases of sequential data. *IEEE Trans. on Signal Processing* to appear.

[51] J. Nocedal and S.J. Wright (2006). *Numerical Optimization, Second Edition*, Springer, New York.

[52] J. Paisley and L. Carin (2009). Hidden Markov models with stick breaking priors, to appear *IEEE Trans. on Sig. Proc.*

[53] J. Paisley and L. Carin (2009). Dirichlet process mixture models with multiple modalities. *Proc. of ICASSP*, pp. 1613-1616.

[54] J. Paisley and L. Carin (2009). Nonparametric factor analysis with beta process priors. *Proc. of ICML*, pp. 777-784.

[55] J. Paisley and L. Carin (2010). A nonparametric Bayesian model for kernel matrix completion. *Proc. of ICASSP*, Dallas, TX.

[56] J. Paisley and L. Carin (2010). Active learning and basis selection for kernel-based linear models: a Bayesian perspective. *IEEE Trans. on Sig. Proc.*, to appear.

[57] J. Paisley and L. Carin (2010). A stick-breaking construction of the beta process. *Proc. of ICML*, submitted.

[58] J. Paisley, M. Zhou, G. Sapiro and L. Carin (2010). Nonparametric image interpolation and dictionary learning using spatially-dependent Dirichlet and beta process priors. *ICIP 2010*, submitted.

[59] Y.C. Pati, R. Rezaiifar, P.S. Krishnaprasad (1993). Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. *Asilomar Conf on Signals Systems and Computers*.

[60] Y. Qi, J. Paisley and L. Carin (2007). Music analysis using hidden Markov mixture models. *IEEE Trans. on Signal Processing*, vol. 55, pp. 5209-5224.

[61] L.R. Rabiner (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE,* Vol. 77, No 2 pp. 257-286.

[62] P. Rai and H. Daumé (2008). The infinite hierarchical factor regression model *NIPS*.

[63] L. Ren, D. Dunson, S. Lindroth and L. Carin (2009) Dynamic nonparametric Bayesian models for analysis of music. *Journal of the American Statistical Association*, to appear.

[64] L. Ren, L. Du, L. Carin and D. Dunson (2010). Logistic stick-breaking process. *Journal of Machine Learning Research* submitted.

[65] A. Rodriquez, D.B. Dunson, and A.E. Gelfand (2008). The nested Dirichlet process. *Journal of the American Statistical Association,* to appear.

[66] A. Rodriguez and D.B. Dunson (2010). Nonparametric Bayesian models through probit stick-breaking processes. *Univ. California Santa Cruz Technical Report*.

[67] N.A. Rosenberg, J.K. Pritchard, J.L. Weber, H.M. Cann, K.K. Kidd, L.A. Zhivotovsky and M.W. Feldman (2002). Genetic structure of human populations. *Science*, vol. 298, pp. 2381-2385.

[68] J. Sethuraman (1994). A constructive definition of Dirichlet priors. *Statistica Sinica,* 4:639-650.

[69] S. Shringarpure and E.P. Xing (2008). mStruct: A New Admixture Model for Inference of Population Structure in Light of Both Genetic Admixing and Allele Mutation. *Proceedings of the 25th ICML*, pp. 952-959.

[70] Y.W. Teh, M.I. Jordan, M.J. Beal and D.M. Blei (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association,* 101(476):1566-1581.

[71] Y.W. Teh and D. Görür and Z. Ghahramani (2007). Stick-breaking construction for the indian buffet process. *AISTATS*.

[72] Y.W. Teh and D. Görür (2009). Indian buffet processes with power-law behavior. *NIPS*.

[73] R. Thibaux and M.I. Jordan (2007). Hierarchical beta processes and the Indian buffet process. *AISTAT 2007*.

[74] R. Tibshirani (1994). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, vol. 58.

[75] M. Tipping (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, vol. 1.

[76] M. West (2003). Bayesian Factor Regression Models in the "Large p, Small n" Paradigm. *Bayesian Statistics*.

[77] J. Winn (2004). *Variational Message Passing and its Applications*. PhD Thesis, Inference Group, Cavendish Laboratory, University of Cambridge.

[78] S.S. Wilks (1962). *Mathematical Statistics*. John Wiley, New York.

[79] R.L. Wolpert and K. Ickstadt (1998). Simulations of Lévy random fields. *Practical and Semiparametric Bayesian Statistics*, pp. 227-242.

[80] Y. Xue, X. Liao, L. Carin and B. Krishnapuram (2007). Multi-task learning for classification with Dirichlet process priors. *The Journal of Machine Learning Research*, vol. 8, pp. 35-63.

[81] A. Zaas and M. Chen and J. Varkey and T. Veldman and A.O. Hero and J. Lucas and Y. Huang and R. Turner and A. Gilbert and R. Lambkin-Williams and N.C. Oien and B. Nicholson and S. Kingsmore and L. Carin and C.W. Woods and G.S. Ginsburg (2009). Gene Expression Signatures Diagnose Influenza and Other Symptomatic Respiratory Viral Infections in Humans *Cell Host & Microbe* vol. 6, pp. 207-217.

[82] M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro and L. Carin (2009). Non-parametric Bayesian dictionary learning for sparse image representations. *NIPS 2009*.

Biography

John Paisley was born October 14, 1981 in Milwaukee, Wisconsin. He graduated from Marquette University High School, Milwaukee, WI, in 2000. He obtained his B.S.E. from Duke University in 2004, majoring in Electrical & Computer Engineering and Computer Science, and minoring in Classical Studies. He obtained his M.S. in Electrical & Computer Engineering from Duke University in 2007. His thesis title was: "Machine Learning Applications in Music Recommendation." His research focuses on Bayesian statistical models for machine learning.