Bayesian Analysis (2014) **9**, Number 2, pp. 259–286

Matrix-Variate Dirichlet Process Priors with Applications

Zhihua Zhang * and Dakan Wang † and Guang Dai ‡ and Michael I. Jordan §

Abstract. In this paper we propose a matrix-variate Dirichlet process (MATDP) for modeling the joint prior of a set of random matrices. Our approach is able to share statistical strength among regression coefficient matrices due to the clustering property of the Dirichlet process. Moreover, since the base probability measure is defined as a matrix-variate distribution, the dependence among the elements of each random matrix is described via the matrix-variate distribution. We apply MATDP to multivariate supervised learning problems. In particular, we devise a nonparametric discriminative model and a nonparametric latent factor model. The interest is in considering correlations both across response variables (or covariates) and across response vectors. We derive Markov chain Monte Carlo algorithms for posterior inference and prediction, and illustrate the application of the models to multivariate regression, multi-class classification and multi-label prediction problems.

Keywords: Dirichlet processes, nonparametric dependent modeling, matrix-variate distributions, nonparametric discriminative analysis, latent factor regression

1 Introduction

Given a set of observed data pairs, $\{(\mathbf{x}_i, \mathbf{y}_i)\}_1^n$, classical multiple regression aims to model the dependency between $\mathbf{x}_i$ and $\mathbf{y}_i$. In an increasingly broad class of problem domains it is desirable to capture additional dependencies in such paired data sets, in particular dependence among the $\{\mathbf{x}_i\}$ (which induces dependence among the $\{\mathbf{y}_i\}$), and dependence among the components of the vectors $\mathbf{y}_i$ (Ibrahim and Kleinman 1998; Gelfand et al. 2005; Xue et al. 2007; Dunson et al. 2007). The latter dependency is particularly important in the setting of classification (where the components of $\mathbf{y}_i$ are binary); a variety of so-called multi-class and multi-label classification problems involve dependencies among these components (Caruana 1997; Tewari and Bartlett 2007).

Bayesian nonparametric models have shown promise in treating general classes of dependencies such as these, with the dependent Dirichlet process (DDP) of MacEachern (1999) providing a flexible general framework for Bayesian nonparametric model specification in which dependencies are captured via dependent collections of random mea-

*MOE-Microsoft Key Lab for Intelligent Computing and Intelligent Systems, Department of Computer Science & Engineering, Shanghai Jiao Tong University, Shanghai, China, zhihua@sjtu.edu.cn

†Twitter Inc, San Francisco, USA, fightiori@gmail.com

‡Department of Computer Science & Engineering, Shanghai Jiao Tong University, Shanghai, China, guang.gdai@gmail.com

§Computer Science Division and Department of Statistics, University of California, Berkeley, USA, jordan@eecs.berkeley.edu

 DOI:10.1214/13-BA853

sures. Other methods based on dependent stochastic processes include Gelfand et al. (2005), who developed a spatial DP model in which the base distribution is defined as a Gaussian process. This spatial DP can be regarded as a "single-p" DDP (MacEachern 2000).

While these general nonparametric frameworks provide the requisite flexibility, they can be challenging to deploy in practice, particularly in the setting of large-scale data, due to the complex procedures that are generally required for posterior inference. For example, the spatial DP model typically requires repeatedly inverting $n \times n$ matrices or computing the determinant of $n \times n$ matrices, which limits the efficient application of the model in large-scale datasets.

In the current paper, we explore a simpler approach; namely, we use a classical Dirichlet process (DP) mixture (Antoniak 1974; Ferguson 1973), but with a base measure that is a matrix-variate distribution. We refer to the resulting prior as a *matrix-variate DP* (MATDP). The nonparametric component of the model is a DP mixture, and the MATDP can be viewed as a "single-p" DDP. Thus, we can proceed via a straightforward application of well-established Markov chain Monte Carlo (MCMC) techniques (Bush and MacEachern 1996; Escobar and West 1995; MacEachern 1998; Neal 2000), capturing the two kinds of dependencies referred to above with a model that is relatively easy to fit in practice. A particular advantage of our approach over the spatial DP is the computational efficiency.

Our focus in this article is the use of the MATDP prior in latent factor analysis, building on the Bayesian latent factor regression model of West (2003). In the latter model, which is geared to high-dimensional problems, it is assumed that $\mathbf{x}$ and $\mathbf{y}$ follow latent factor models, with a connection between $\mathbf{x}$ and $\mathbf{y}$ implemented via the sharing of a common latent vector. We place MATDP priors on the loading matrices for the two latent factor analyzers and thereby obtain a flexible Bayesian prior for high-dimensional $\mathbf{x}$ and $\mathbf{y}$. The overall model is a DP-based Latent Factor Model (DP-LFM).

Our DP-LFM can be viewed as finding a low-dimensional latent space and implementing a regression on the latent subspace, and can thus be regarded as an approach to jointly carry out dimensionality reduction and regression. This highlights an advantage of DP-LFM over some related models that separate these processes, in particular the Dirichlet process multinomial logit (dpMNL) model of Shahbaba and Neal (2009) and the DP-generalized linear model (DP-GLM) of Hannah et al. (2010). However, our DP-LFM retains the nonlinear aspect of the dpMNL. Within each component of the MATDP mixture, the relationship between $\mathbf{y}$ and $\mathbf{x}$ (i.e., $p(\mathbf{y}|\mathbf{x})$) is expressed using a (generalized) linear function. The overall relationship becomes piecewise linear because the mixture typically contains many components. Thus, the overall model is essentially nonlinear.

We also show how to extend DP-LFM to classification problems, in particular multi-class and multi-label prediction. We note that nonparametric latent factor models have been studied in this setting by Rai and Daumé III (2009, 2010), who proposed an infinite canonical component analysis (CCA) model based on the Indian buffet process (Griffiths and Ghahramani 2005). In fact, our DP-LFM can be also regarded as a nonparametric

extension of probabilistic CCA, but in our case we build on DP mixtures.

Although our focus is latent factor analysis, as a stepping stone we also present a model in which the MATDP is used as a prior for a multinomial probit regression model. Our model is discriminative (Ng and Jordan 2002), because it estimates the conditional distribution $p(\mathbf{y}|\mathbf{x})$, but not the distribution of covariates, $p(\mathbf{x})$. In contrast, the dpMNL and DP-LFM models are generative.

The remainder of the paper is organized as follows. Section 2 overviews our notation and Section 3 presents the matrix-variate DP model. We discuss an application to multinomial probit regression in Section 4 and a nonparametric latent factor model in Section 5. Experimental analyses are presented in Section 6. Finally, we conclude our work in Section 7.

2 Notation

We let $\mathbf{0}$ represent the zero vector (or matrix) whose dimensionality is dependent upon the context, $\mathbf{1}_m$ be the $m\times 1$ vector of ones, and $\mathbf{I}_m$ denote the $m\times m$ identity matrix. Let $\mathrm{Ga}(\alpha|a_\alpha, b_\alpha)$ denote that positive random variable α follows a Gamma distribution with shape parameter a_α and scale parameter b_α, and $G \sim \mathrm{DP}(\alpha G_0)$ denote that random measure G follows a DP prior with base probability measure G_0 and concentration parameter $\alpha > 0$. We employ the notation of Gupta and Nagar (2000) for matrix-variate distributions. That is, for a $p\times q$ random matrix $\mathbf{Y}$, $\mathbf{Y} \sim N_{p,q}(\cdot|\mathbf{M}, \mathbf{A}\otimes\mathbf{B})$ means that $\mathbf{Y}$ follows a matrix-variate normal distribution with mean matrix $\mathbf{M}$ $(p\times q)$ and covariance matrix $\mathbf{A}\otimes\mathbf{B}$, where $\mathbf{A}$ $(p\times p)$, $\mathbf{B}$ $(q\times q)$ are positive definite, and $\mathbf{A}\otimes\mathbf{B}$ is the Kronecker product of $\mathbf{A}$ and $\mathbf{B}$. Additionally, for an $s\times s$ random matrix $\mathbf{C}$, let $\mathbf{C} \sim W_s(\cdot|r, \mathbf{D})$ denote that $\mathbf{C}$ follows a Wishart distribution with r $(\geq s)$ degrees of freedom and an $s\times s$ positive definite parameter matrix $\mathbf{D}$. Finally, we let $\mathbf{x} \in \mathbb{R}^p$ and $\mathbf{y} \in \mathbb{R}^q$ for the covariate vector and response vector, respectively. We also use the terminology "input vector" for the covariate vector.

3 Matrix-variate DP Priors

In a conventional Dirichlet process mixture (DPM) model, one assumes that the observations $\mathbf{z}_i$, for $i = 1, \ldots, n$, are drawn from a mixture component parameterized by $\boldsymbol{\theta}_i \in \Theta$. Furthermore, the $\boldsymbol{\theta}_i$'s are generated by the distribution G, which is in turn assumed to follow the DP prior $\mathrm{DP}(\alpha G_0)$.

In this paper we are concerned with the case that the parameters are a set of $p\times q$ random matrices $\boldsymbol{\Theta}_i$. To capture relationships among the $\boldsymbol{\Theta}_i$'s, we introduce a DP prior to model the joint distribution of the $\boldsymbol{\Theta}_i$'s. That is,

$$[\boldsymbol{\Theta}_i|G] \overset{iid}{\sim} G, \quad i = 1, \ldots, n,$$
$$G \sim \mathrm{DP}(\alpha G_0).$$

We assume that the base probability measure G_0 follows a matrix-variate distribution. We thus refer to the resulting DP as a *matrix-variate DP* (MATDP). Please also see Zhang et al. (2010) for an earlier use of the MATDP prior in the setting of linear regression.

As in the case of the conventional DP prior (Blackwell and MacQueen 1973), integrating over G yields a Pólya urn scheme for the $\mathbf{\Theta}_i$'s; that is,

$$\mathbf{\Theta}_1 \sim G_0,$$
$$[\mathbf{\Theta}_i|\mathbf{\Theta}_1, \ldots, \mathbf{\Theta}_{i-1}] \sim \frac{\alpha G_0 + \sum_{l=1}^{i-1} \delta_{\mathbf{\Theta}_l}}{\alpha + i - 1},$$

where $\delta_{\mathbf{\Theta}_l}$ is a point mass at $\mathbf{\Theta}_l$. Obviously, as $\alpha \to 0$ all the $\mathbf{\Theta}_i$'s are identical to $\mathbf{\Theta}_1$, which follows G_0. The $\mathbf{\Theta}_i$'s are drawn iid from G_0 when $\alpha \to \infty$.

The Pólya urn representation of the marginals of the random distribution G leads to the well-known clustering property of the DP, which plays a central role in Bayesian nonparametric inference and computation. Assume that there are c distinct values among the $\mathbf{\Theta}_i$'s, denoted $\Phi = \{\mathbf{\Phi}_1, \ldots, \mathbf{\Phi}_c\}$, and assume that there are n_k occurrences of $\mathbf{\Phi}_k$ such that $\sum_{k=1}^{c} n_k = n$. The vector of configuration indicators, $\mathbf{w} = (w_1, \ldots, w_n)$, is defined by $w_i = k$ if and only if $\mathbf{\Theta}_i = \mathbf{\Phi}_k$ for $i = 1, \ldots, n$. Thus $(\Phi, \mathbf{w})$ is an equivalent representation of $\mathbf{\Theta}$. Considering that the $\mathbf{\Theta}_i$'s are exchangeable, we rewrite the Pólya urn scheme as

$$[\mathbf{\Theta}_i|\mathbf{\Theta}_{-i}] \sim \frac{\alpha G_0 + \sum_{k=1}^{c} n_{k(-i)} \delta_{\mathbf{\Phi}_k}}{\alpha + n - 1}$$

and

$$\mathbf{\Phi}_k \overset{iid}{\sim} G_0, \ k = 1, \ldots, c.$$

Here $\mathbf{\Theta}_{-i}$ represents $\{\mathbf{\Theta}_l : l \neq i\}$ and $n_{k(-i)}$ refers to the cardinality of cluster k, with $\mathbf{\Theta}_i$ removed.

In the MATDP mixture specification, we accordingly have

$$\begin{aligned} [\mathbf{z}_i|\mathbf{\Theta}_i] &\overset{ind}{\sim} F(\mathbf{z}_i|\mathbf{\Theta}_i), \\ [\mathbf{\Theta}_i|G] &\overset{iid}{\sim} G, \\ G &\sim \mathrm{DP}(\alpha G_0). \end{aligned}$$

As a concrete example, let G_0 follow a matrix-variate normal distribution of the form

$$G_0(\cdot|\mathbf{\Sigma}, \mathbf{\Lambda}) = N_{p,q}(\cdot|\mathbf{M}, \ \mathbf{A}\otimes\mathbf{B}).$$

It is worth emphasizing that the dependence between the $\mathbf{\Theta}_i$'s is characterized by the DP prior, while the dependence among the elements of each matrix $\mathbf{\Theta}_i$ is represented by the covariance matrix $\mathbf{A}\otimes\mathbf{B}$. This prior can be regarded as a specific instance of a single-p dependent Dirichlet prior (MacEachern 2000). In Sections 4 and 5, we illustrate the application of this MATDP prior in multi-class discriminant models and latent factor models, respectively.

4 Multinomial Probit Regression via MATDP Mixing

In this section we present an application of the MATDP prior to the multi-class (or polychotomous) classification problem. We consider a q-class classification problem in which the training dataset is $\{(\mathbf{x}_i, \mathbf{y}_i)\}_1^n$, with covariates $\mathbf{x}_i \in \mathbb{R}^p$ and response vectors $\mathbf{y}_i \in \{0,1\}^q$. Note that $\mathbf{y}_i = (y_{i1}, \ldots, y_{iq})'$ is the multinomial indicator vector with elements $y_{ij} = 1$ if $\mathbf{x}_i$ belongs to the jth class and $y_{ij} = 0$ otherwise.

In order to facilitate the implementation of Bayesian inference, we employ a classical data augmentation technique to deal with non-Gaussian distributions (Albert and Chib 1993; Holmes and Held 2006). We define $\{\mathbf{z}_1, \ldots, \mathbf{z}_n\} \subset \mathbb{R}^q$ as a set of auxiliary vectors, and relate $\mathbf{y}_i = (y_{i1}, \ldots, y_{iq})'$ to $\mathbf{z}_i = (z_{i1}, \ldots, z_{iq})'$ through the probit link (Denison et al. 2002) due to its tractability in Bayesian inference.

Additionally, we consider a set of regression functions, $\{f_j(\mathbf{x})\}$, defined as linear combinations of m basis functions $\{g_l(\mathbf{x})\}$; that is,

$$f_j(\mathbf{x}) = b_{j0} + \sum_{l=1}^{m} b_{jl} g_l(\mathbf{x}), \quad j = 1, \ldots, q,$$

where the b_{j0}'s are offset terms and the b_{jl}'s are regression coefficients. An important and popular choice for the basis function is $g_l(\mathbf{x}) = K(\mathbf{x}_l, \mathbf{x})$ where $K(\cdot,\cdot)$ is a reproducing kernel function (Schölkopf and Smola 2002). We will employ this choice in this paper due to its ability to capture nonlinear relationships. In this case, we have $m = n$.

Let $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_q]$ where $\mathbf{b}_j = (b_{j0}, b_{j1}, \ldots, b_{jm})'$ for $j = 1, \ldots, q$, and $\mathbf{g}(\mathbf{x}) = (1, g_1(\mathbf{x}), \ldots, g_m(\mathbf{x}))'$. We define the following regression model:

$$\mathbf{z}_i = \mathbf{B}'\mathbf{g}_i + \epsilon_i,$$

where $\mathbf{g}_i = \mathbf{g}(\mathbf{x}_i)$ for short and ϵ_i is a Gaussian error. We aim to capture the dependence among the response variables and among the data samples. To take a Bayesian nonparametric approach, we allow each $\mathbf{g}_i$ to have its own regression coefficient matrix $\mathbf{B}_i$, placing a MATDP prior on the joint distribution of the $\mathbf{B}_i$.

In summary, we have

$$\begin{aligned} y_{ij} &= \begin{cases} 1 & \text{if } j = \operatorname{argmax}_{1\le k\le q}\{z_{ik}\}, \\ 0 & \text{otherwise,} \end{cases} \qquad i = 1, \ldots, n \\ [\mathbf{z}_i | \mathbf{B}_i, \boldsymbol{\Sigma}] &\overset{ind}{\sim} N_q(\mathbf{z}_i | \mathbf{B}_i'\mathbf{g}_i,\ \boldsymbol{\Sigma}), \quad i = 1, \ldots, n, \\ [\mathbf{B}_i | G] &\overset{iid}{\sim} G, \quad i = 1, \ldots, n, \\ G &\sim \mathrm{DP}(\alpha G_0). \end{aligned} \tag{1}$$

Furthermore, we define G_0 as

$$G_0(\cdot | \boldsymbol{\Sigma}, \boldsymbol{\Lambda}) = N_{m+1,q}(\cdot | \mathbf{0},\ \boldsymbol{\Lambda} \otimes \boldsymbol{\Sigma}).$$

Here $\mathbf{\Sigma}$ is a $q\times q$ positive semidefinite matrix. To make the model identifiable, one typically imposes the constraint that $\sum_{j=1}^{q} z_{ij} = 0$ for $i = 1, \ldots, n$. This implies that the variates of $\mathbf{z}_i$ are mutually dependent. We impose the constraint via the use of a singular normal distribution for $\mathbf{z}_i$ (Mardia et al. 1979). In particular, we assume that $\mathbf{\Sigma}$ is of rank $q-1$ and satisfies the condition $\mathbf{\Sigma}\mathbf{1}_q = 0$. In this case, we can write $\mathbf{\Sigma} = \begin{bmatrix} \mathbf{I}_{q-1} \\ -\mathbf{1}'_{q-1} \end{bmatrix} \mathbf{\Sigma}_{11}[\mathbf{I}_{q-1}, -\mathbf{1}_{q-1}]$ where $\mathbf{\Sigma}_{11}$ is a $(q-1)\times(q-1)$ positive definite matrix. Since the Moore-Penrose pseudoinverse $\mathbf{\Sigma}^{+}$ of $\mathbf{\Sigma}$ is

$$\mathbf{\Sigma}^{+} = \mathbf{H}_q \mathbf{\Sigma}_{11}^{-1} \mathbf{H}'_q,$$

where $\mathbf{H}_q$ $(q\times(q-1))$ contains the first $q-1$ columns of the centering matrix $\mathbf{C}_q = \mathbf{I}_q - \frac{1}{q}\mathbf{1}_q\mathbf{1}'_q$, the conditional density of $\mathbf{z}_i$ on $\mathbf{B}_i$ is

$$\frac{(2\pi)^{-\frac{q-1}{2}}}{q^{1/2}|\mathbf{\Sigma}_{11}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{z}_i - \mathbf{B}'_i\mathbf{g}_i)'\mathbf{H}_q\mathbf{\Sigma}_{11}^{-1}\mathbf{H}'_q(\mathbf{z}_i - \mathbf{B}'_i\mathbf{g}_i)\right).$$

Considering that the rows of $\mathbf{B}_i$ are associated with the basis functions which are typically independent, we set $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{m+1})$, a diagonal matrix with $\lambda_i > 0$ for $i = 1, \ldots, m+1$. We will see that such a setting can make computations efficient. In addition, we assume that α and λ_i^{-1} follow Gamma distributions: $\mathrm{Ga}(\alpha|a_\alpha,\ b_\alpha)$ and $\mathrm{Ga}(\lambda_i^{-1}|\frac{a_i}{2},\ \frac{b_i}{2})$; and we assume that $\mathbf{\Sigma}_{11}^{-1}$ follows a Wishart distribution: $W_{q-1}(\mathbf{\Sigma}_{11}^{-1}|\rho,\ \mathbf{R}_{11}^{-1})$.

Since our model directly describes the conditional distribution $p(\mathbf{y}|\mathbf{x})$, it can be regarded as a nonparametric discriminative model. Recall that the relationship between $\mathbf{z}_i$ and $\mathbf{g}_i$ is linear; that is, $\mathbb{E}[\mathbf{z}_i|\mathbf{g}_i] = \mathbf{B}'_i\mathbf{g}_i$. However, distinct pairs $\mathbf{z}_i$ and $\mathbf{g}_i$ possibly correspond to distinct regression coefficient matrices $\mathbf{B}_i$, which implies that the overall relationship is piecewise linear. Thus, the nonparametric specification for $\mathbf{B}_i$ makes the resulting model nonlinear.

Finally, the clustering property of DPs mentioned in Section 3 naturally allows the sharing of statistical strength between the covariate vectors and between the response variables. Moreover, the clustering property is able to transfer statistical strength from existing regression coefficient matrices to new regression coefficient matrices (see equation (4)), and thus yield out-of-sample prediction as will be discussed in more detail in the following section.

4.1 Posterior Sampling and Prediction

We now devise a posterior sampling MCMC algorithm for our model. Posterior sampling is built on the Pólya urn scheme of the DP so as to take advantage of the clustering property.

Using the same notation as in Section 3, we have

$$[\mathbf{B}_i|\mathbf{B}_{-i}] \sim \frac{\alpha N_{m+1,q}\Big(\cdot\,|\mathbf{0},\ \mathbf{\Lambda}\otimes\mathbf{\Sigma}\Big) + \sum_{k=1}^{c} n_{k(-i)}\delta_{\mathbf{Q}_k}}{\alpha+n-1}, \tag{2}$$

where $\mathcal{Q}=\{\mathbf{Q}_1,\ldots,\mathbf{Q}_c\}$ includes c distinct values among the $\mathbf{B}_i$, and

$$\mathbf{Q}_k \overset{iid}{\sim} N_{m+1,q}\Big(\mathbf{Q}_k|\mathbf{0},\ \mathbf{\Lambda}\otimes\mathbf{\Sigma}\Big),\ k=1,\ldots,c.$$

Consequently, we can express the joint distribution of $\mathbf{Z}=[\mathbf{z}_1,\ldots,\mathbf{z}_n]'$ $(n\times q)$ as

$$[\mathbf{Z}|\mathbf{w},\mathcal{Q}] \sim \prod_{k=1}^{c}\prod_{i:\ w_i=k} N_q(\mathbf{z}_i|\mathbf{Q}_k'\mathbf{g}_i,\ \mathbf{\Sigma}).$$

Integrating out the $\mathbf{Q}_k$ yields the conditional (on $\mathbf{w}$) marginal distribution of $\mathbf{Z}$ as

$$[\mathbf{Z}|\mathbf{w},\mathbf{\Lambda},\mathbf{\Sigma}] \sim \prod_{k=1}^{c} N_{n_k,q}\Big(\mathbf{Z}_k|\mathbf{0},\ (\mathbf{I}_{n_k}+\mathbf{G}_k\mathbf{\Lambda}\mathbf{G}_k')\otimes\mathbf{\Sigma}\Big),$$

where $\mathbf{Z}_k$ and $\mathbf{G}_k$ are respectively $n_k\times q$ and $n_k\times(m+1)$ matrices consisting of those $\mathbf{z}_i$ and $\mathbf{g}_i$ with $w_i=k$. For each $k=1,\ldots,c$, we have

$$[\mathbf{Q}_k|\mathbf{Z},\mathbf{w},\mathbf{\Lambda},\mathbf{\Sigma}] \sim N_{m+1,q}\Big(\mathbf{Q}_k|\mathbf{\Theta}_k\mathbf{G}_k'\mathbf{Y}_k,\ \mathbf{\Theta}_k\otimes\mathbf{\Sigma}\Big), \tag{3}$$

where $\mathbf{\Theta}_k=(\mathbf{\Lambda}^{-1}+\mathbf{G}_k'\mathbf{G}_k)^{-1}$.

Since given $\mathbf{Z}$, $\mathbf{Y}=[\mathbf{y}_1,\ldots,\mathbf{y}_n]'$ $(n\times q)$ is independent of the other model parameters, posterior sampling is achieved by generating realizations of the parameters from the conditional joint density $[\{\mathbf{B}_i\}_{i=1}^n,\mathbf{\Lambda},\mathbf{\Sigma}|\mathbf{Z}]$ (see Appendix I for a detailed presentation). As for the estimate of $\mathbf{Z}$, we only need to insert a step of updating $\mathbf{Z}$ from $p(\mathbf{Z}|\mathbf{Y},\{\mathbf{B}_i\}_{i=1}^n,\mathbf{\Sigma})$ into the MCMC algorithm in Appendix I. To estimate $\mathbf{z}_i$, we first sample an auxiliary vector $\mathbf{s}_i=(s_{i1},\ldots,s_{i,q-1})$ from the truncated normal; that is, $[s_{ij}|\mathbf{g}_i,y_{ij}] \sim N(s_{ij}|\boldsymbol{\sigma}_j\mathbf{B}_i'\mathbf{g}_i,1)$ subject to $s_{ij}>\max(\max_{l\neq j}\{s_{il}\},0)$ if $y_{ij}=1$ when $j=1,\ldots,q-1$, and $[s_{ij}|\mathbf{g}_i,y_{ij}] \sim N(s_{ij}|\boldsymbol{\sigma}_j\mathbf{B}_i'\mathbf{g}_i,1)$ subject to $s_{ij}<0$ if $y_{iq}=1$. We then let $\mathbf{z}_i=\mathbf{H}_q\mathbf{\Sigma}_{11}^{1/2}\mathbf{s}_i$. Here $\boldsymbol{\sigma}_j$ is the jth row of $\mathbf{\Sigma}_{11}^{-1/2}\mathbf{H}_q'$.

Our method groups the regression coefficient matrices $\mathbf{B}_i$ into c clusters by using the MATDP prior. The main computational burden of our method comes from the calculation of $\mathbf{\Theta}_k$, but fortunately we can use the Sherman-Morrison-Woodbury formula (Golub and Loan 1996) to calculate $\mathbf{\Theta}_k$ efficiently. In particular, we have

$$\mathbf{\Theta}_k=(\mathbf{\Lambda}^{-1}+\mathbf{G}_k'\mathbf{G}_k)^{-1}=\mathbf{\Lambda}-\mathbf{\Lambda}\mathbf{G}_k'(\mathbf{I}_{n_k}+\mathbf{G}_k\mathbf{\Lambda}\mathbf{G}_k')^{-1}\mathbf{G}_k\mathbf{\Lambda}.$$

Thus, the above formula allows us to invert an $n_k\times n_k$ matrix instead of an $n\times n$ matrix when the basis functions $g_j(\mathbf{x})$ are defined as the kernel function $K(\mathbf{x}_j,\mathbf{x})$ for $j=$

$1, \ldots, n$ (see Appendix I). Since n_k is typically far smaller than n, the algorithm can be efficient for a large-scale dataset.

Given a new input vector $\mathbf{x}_0$, let us now consider how to predict its label $\mathbf{y}_0 \in \{0,1\}^q$. Assume $\mathbf{B}_0$ and $\mathbf{z}_0$ are the coefficient matrix and the auxiliary vector associated with $\mathbf{y}_0 = (y_{01}, \ldots, y_{0q})'$. We have

$$[\mathbf{B}_0|\mathcal{Q}, \mathbf{w}, \alpha, \mathbf{\Lambda}] \sim \frac{\alpha}{\alpha+n} N_{m+1,q}(\cdot|\mathbf{0},\ \mathbf{\Lambda}\otimes\mathbf{\Sigma}) + \frac{1}{\alpha+n}\sum_{k=1}^{c} n_k \delta_{\mathbf{Q}_k}, \tag{4}$$

which yields out-of-sample prediction. The posterior distribution of $\mathbf{y}_0$ is given by

$$p(\mathbf{y}_0|\mathbf{x}_0, \mathbf{Y}) = \int p(\mathbf{y}_0|\mathbf{x}_0, \mathbf{B}_0) p(\mathbf{B}_0|\mathbf{Q}) p(\mathbf{Q}|\mathbf{Y}) d\mathbf{Q} d\mathbf{B}_0.$$

To approximate this integral via Monte Carlo integration, we draw $\mathbf{B}_0^{(t)}$ from equation (4) and then compute

$$\hat{p}(y_{0l} = 1|\mathbf{x}_0, \mathbf{Y}) = \frac{1}{T}\sum_{t=1}^{T} p\left(y_{0l} = 1|\mathbf{x}_0, \mathbf{B}_0^{(t)}, \Omega^{(t)}\right),$$

where the $\Omega^{(t)}$ are the MCMC realizations of all the model parameters (after the burn-in) but the $\mathbf{Q}_k$.

4.2 Related Work

In related work, Ibrahim and Kleinman (1998) and Xue et al. (2007) suggested assigning a DP prior to the columns (denoted $\mathbf{b}_j$) of $\mathbf{B} \in \mathbb{R}^{p\times q}$. Relative to the hierarchical model in equation (1), the model of Ibrahim and Kleinman (1998) and Xue et al. (2007) for the multivariate generalized linear regression problem is specified as

$$\begin{aligned} [\mathbf{y}_i|\mathbf{B}, \mathbf{x}_i] &\overset{ind}{\sim} F(\mathbf{y}_i|\mathbf{B}'\mathbf{x}_i), \quad i = 1, \ldots, n, \\ [\mathbf{b}_j|G] &\overset{iid}{\sim} G, \quad j = 1, \ldots, q, \\ G &\sim \mathrm{DP}(\alpha G_0). \end{aligned} \tag{5}$$

This model is able to capture the dependence among the response variables but ignores the dependence among the covariate vectors. Moreover, since the dimensionality q of the response is usually not too large in practical applications, the clustering property of DPs might place all of the columns in a single class, enforcing too much sharing (Bush et al. 2010). Thus, it is necessary to take a larger mass parameter in practice. It is worth pointing out that the limiting case of the model in equation (5) at $\alpha = \infty$ is identical to the limiting case of the corresponding MATDP model at $\alpha = 0$.

Alternatively, Gelfand et al. (2005) proposed the spatial DP (sDP) model, which is

$$\begin{aligned}
[\mathbf{y}_{\cdot j}|\mathbf{s}_j, \sigma^2] &\overset{ind}{\sim} N_n(\mathbf{y}_{\cdot j}|\mathbf{s}_j,\ \sigma^2\mathbf{I}_n), \quad j = 1, \ldots, q, \\
[\mathbf{s}_j|G] &\overset{iid}{\sim} G, \quad j = 1, \ldots, q, \\
G &\sim \mathrm{DP}(\alpha G_0), \\
G_0(\cdot|\mathbf{K}, \tau) &= N_n(\cdot|\mathbf{0},\ \tau^{-1}\mathbf{K}),
\end{aligned}$$

where $\mathbf{y}_{\cdot j} = (y_{1j}, \ldots, y_{nj})'$ and $\mathbf{K} = [K(\mathbf{x}_i, \mathbf{x}_j)]$ is the $n \times n$ kernel matrix. We see that the base distribution in the sDP is defined as a Gaussian process (GP). Specifically, this model describes the dependence among the response variables via a DP, and the dependence among the samples via a GP. A difficulty with this approach is that the MCMC algorithm for the sDP involves the computation of $n \times n$ matrices at each sweep; in particular, the algorithm needs to calculate the densities of n-variate normal distributions in obtaining the posterior distribution $p(\mathbf{s}_j|\mathbf{s}_{-j}, \mathbf{Y})$. This n^3 computational complexity limits the applicability of the sDP model for large-scale datasets.

Figure 1 provides a graphical representation of all three of these three models in the setting of regression.

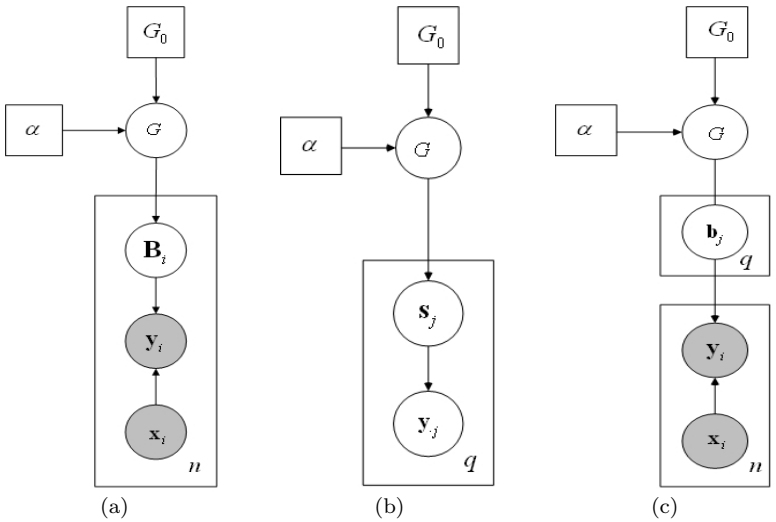


Figure 1: Graphical representations under regression setting: (a) MATDP, (b) sDP, and (c) the model defined in equation (5) (called DPC in Section 6.1).

Another example of related work is the kernel weighted mixture of DPs (Dunson et al. 2007), which is able to capture the relationship among the covariate vectors. However, it does not capture dependence among the response variables. Our approach is also different from the method of Dunson et al. (2008) in which only one regression coefficient matrix is employed for all samples and a so-called matrix stick-breaking process is proposed to define a joint prior for the elements of this regression coefficient matrix.

5 Nonparametric Latent Factor Models

We turn to our proposed framework for Bayesian nonparametric latent factor analysis, with application to classification problems in which $\mathbf{y}_i \in \{0,1\}^q$. We build on the latent factor analysis model of West (2003), which has the following specification:

$$\begin{aligned}\mathbf{x}_i &= \mathbf{A}\boldsymbol{\eta}_i + \boldsymbol{\mu} + \boldsymbol{\epsilon}_i,\\ \boldsymbol{\eta}_i &\sim N_r(\cdot|\mathbf{0}, \mathbf{I}_r),\\ \boldsymbol{\epsilon}_i &\sim N_p(\cdot|\mathbf{0}, \boldsymbol{\Sigma}),\end{aligned}$$

where $\boldsymbol{\eta}_i$ is a r-dimensional vector of latent factors, $\boldsymbol{\mu}$ is a p-dimensional offset term and $\mathbf{A}$ is a $p \times r$ matrix of factor loadings. The corresponding response $\mathbf{y}_i$ is obtained from a coupled latent factor analysis model:

$$\begin{aligned}\mathbf{y}_i &= \mathbf{B}\boldsymbol{\eta}_i + \boldsymbol{\nu} + \boldsymbol{\varepsilon}_i,\\ \boldsymbol{\varepsilon}_i &\sim F(\cdot|\mathbf{0},\ \boldsymbol{\Lambda}),\end{aligned}$$

where $\mathbf{B}$ is a $q \times r$ matrix of factor loadings, $\boldsymbol{\nu}$ is a q-dimensional offset term and $F(\cdot)$ can be defined by an exponential family distribution, e.g., Gaussian or multinomial. We will assume a Gaussian distribution in the following presentation, because of our use of Gaussian-based data augmentation techniques (see Section 4).

As we see, the latent factor model of West (2003) connects $\mathbf{x}_i$ and $\mathbf{y}_i$ through the latent vector $\boldsymbol{\eta}_i$. Moreover, the original input $\mathbf{x}_i$ does not enter the model directly; that is, $\mathbf{y}_i$ is conditionally independent of $\mathbf{x}_i$ given $\boldsymbol{\eta}_i$. Typically, r is less than p. Thus, the model directly addresses both dimensionality reduction and regression. When the $F(\boldsymbol{\varepsilon}_i|\mathbf{0},\ \boldsymbol{\Lambda})$ are Gaussian, West (2003) showed that the conditional distribution for $\mathbf{y}_i$ given only $\mathbf{x}_i$ and the model parameters is still Gaussian. This implies that the relationship between $\mathbf{y}_i$ and $\mathbf{x}_i$ is linear.

Carvalho et al. (2010) extended the work of West (2003) by incorporating additional latent factors for responses. To relax the Gaussian assumptions for the latent factors, they used a DP prior to describe the joint distribution of the extended latent factors. We now turn to a new nonparametric extension of the model of West (2003) which preserves its virtues for high-dimensional data while also addressing the issue of potential dependency among the data samples and among the components of the covariate or response vectors, and also capturing nonlinear relationships between the covariates and response variables.

5.1 The Model

Our framework extends the latent factor analysis model of West (2003) to incorporate a MATDP prior. For $i = 1, \ldots, n$, the specification is

$$\begin{aligned}
\mathbf{x}_i &\sim N_p\Big(\cdot\,|\mathbf{A}_i\boldsymbol{\eta}_i + \boldsymbol{\mu}_i,\ \boldsymbol{\Sigma}_i\Big),\\
\mathbf{y}_i &\sim F(\cdot|\mathbf{B}_i\boldsymbol{\eta}_i + \boldsymbol{\nu}_i,\ \boldsymbol{\Lambda}_i),\\
\boldsymbol{\eta}_i &\sim N_r(\cdot|\mathbf{0}, \mathbf{I}_r)\\
\Big[\boldsymbol{\theta}_i|G\Big] &\sim G,\\
G &\sim DP(\alpha G_0),
\end{aligned}$$

where $\boldsymbol{\Sigma}_i = \text{diag}(\sigma_{i1}^2, \ldots \sigma_{ip}^2)$, $\boldsymbol{\Lambda}_i = \text{diag}(\lambda_{i1}^2, \ldots, \lambda_{iq}^2)$, and $\boldsymbol{\theta}_i = \{\mathbf{A}_i, \mathbf{B}_i, \boldsymbol{\mu}_i, \boldsymbol{\nu}_i, \boldsymbol{\Sigma}_i, \boldsymbol{\Lambda}_i\}$ are the parameters which follow a joint DP prior $\text{DP}(\alpha G_0)$.

The base distribution G_0 over $\boldsymbol{\theta}_i$ is as follows:

$$\begin{aligned}
\mathbf{A}_i &\sim N_{p,r}(\cdot|\mathbf{0},\ \boldsymbol{\Psi}_1 \otimes \boldsymbol{\Phi}_1),\\
\mathbf{B}_i &\sim N_{q,r}(\cdot|\mathbf{0},\ \boldsymbol{\Psi}_2 \otimes \boldsymbol{\Phi}_2),\\
\boldsymbol{\mu}_i &\sim N_p(\cdot|\mathbf{m}_\mu,\ \text{diag}(\mathbf{v}_\mu)^2),\\
\boldsymbol{\nu}_i &\sim N_q(\cdot|\mathbf{m}_\nu,\ \text{diag}(\mathbf{v}_\nu)^2),\\
\log(\sigma_{ij}^2) &\sim N(\cdot|m_{\sigma,j},\ v_{\sigma,j}^2),\\
\log(\lambda_{il}^2) &\sim N(\cdot|m_{\lambda,l},\ v_{\lambda,l}^2).
\end{aligned}$$

Here $v_{\sigma,j}$ represents the jth entry of the vector $\mathbf{v}_\sigma$. The concentration parameter α follows $\log(\alpha^2) \sim N(\cdot|a_\alpha, b_\alpha)$. We further assume that the priors of $m_{\mu,j}$, $v_{\mu,j}^2$, $m_{\sigma,j}$ and $v_{\sigma,j}^2$ are $m_{\mu,j} \sim N(\cdot|0, a_\mu)$, $\log(v_{\mu,j}^2) \sim N(\cdot|0, b_\mu)$, $m_{\sigma,j} \sim N(\cdot|0, a_\sigma)$, and $\log(v_{\sigma,j}^2) \sim N(\cdot|0, b_\sigma)$. The prior for $\boldsymbol{\Psi}_1$ follows the setting for $\boldsymbol{\Lambda}$ in the previous section, but we now assume that $\boldsymbol{\Phi}_1^{-1}$ follows a Wishart distribution $W_p(\cdot|\rho_1, \mathbf{R}_1^{-1})$. Moreover, we suggest that $\rho_1 = p+1$ and $\mathbf{R}_1 = \mathbf{I}_p + \frac{1}{p}\mathbf{1}_p\mathbf{1}_p'$. The setting for $\mathbf{R}_1$ makes the covariates be mutually equicorrelated. Hyperparameters associated with $\mathbf{y}_i$ are defined analogously.

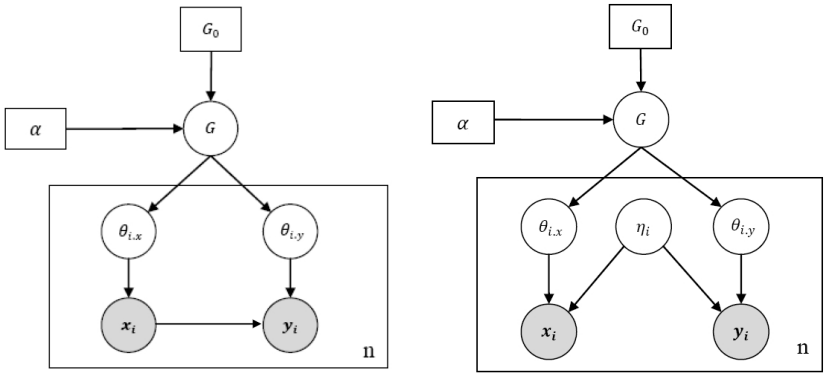
In the above DP-based Latent Factor Model (DP-LFM), the dimensionality of the latent vector $\boldsymbol{\eta}_i$ (i.e., the number of factors) is assumed to be prespecified by practitioners. Although one can potentially assign a sparsity prior for $\mathbf{A}_i$ or $\mathbf{B}_i$ as in Carvalho et al. (2010) to address this issue, we have not investigated such an extension in this paper.

It can be shown that the joint distribution of $(\mathbf{x}_i, \mathbf{y}_i)$ under the DP-LFM follows a mixture-of-Gaussians distribution. In each mixture component, there is a component-specific regressor $\mathbf{B}_i$ responsible for generating the response $\mathbf{y}_i$. Therefore, in different mixture components, covariates and responses are related differently. This piecewise linear relationship implies that the overall model is nonlinear. A related nonlinear model, referred to as dpMNL, has been described by Shahbaba and Neal (2009); we compare the two models graphically in Figure 2. We also note that our model can be viewed as an infinite mixture of factor analyzers, a model which has been considered by Chen et al. (2010) and Görür and Rasmussen (2007). Our work differs in that we

employ a MATDP prior for the loading matrices, allowing us to capture dependencies in these matrices across data samples.

It is worth pointing out that instead of directly relating the input to the response, the factor model introduces a normal latent variable to bridge the input and the response. This brings three benefits for our model over the dpMNL model. First, for high-dimensional inputs, our model transforms the input into a low-dimensional subspace and therefore decreases the complexity of the overall mapping. Second, for inputs with noise, our model denoises the data and therefore makes the training of the estimation of the loading matrices more robust. Third, our model has the capability of accommodating inputs with missing entries.

Finally, to extend the nonparametric specification so that it can handle classification problems, where $\mathbf{y}$ is a label instead of a q-dimensional real vector, we follow the path discussed in Section 4 and assume that $\mathbf{y}$ follows a probit model given $\boldsymbol{\eta}$. We will conduct the empirical analysis of this model in Section 6.3.



(a) dpMNL (b) DP-LFM

Figure 2: Graphical model representations for dpMNL and DP-LFM.

5.2 Posterior Sampling

We devise algorithms for fitting $\{\boldsymbol{\theta}_i\}_{i=1}^n$ and $\{\boldsymbol{\eta}_i\}_{i=1}^n$. Let $\boldsymbol{\theta}_{i,x} = (\mathbf{A}_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ be the parameters associated with $\mathbf{x}_i$, and let $\boldsymbol{\theta}_{i,y}$ define the analogous parameters associated with $\mathbf{y}_i$. In this section, we present a posterior sampling algorithm for $\boldsymbol{\theta}_{i,x}$; note that the sampling algorithm for $\boldsymbol{\theta}_{i,y}$ is a notational variant of that for $\boldsymbol{\theta}_{i,x}$.

Given the discreteness of the random measure G, we assume that there are c distinct values among $\{\boldsymbol{\theta}_i\}_{i=1}^n$, denoted $\{\boldsymbol{\tau}_j = (\mathbf{A}_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \mathbf{B}_j, \boldsymbol{\nu}_j, \boldsymbol{\Lambda}_j)\}_{j=1}^c$. We further introduce n auxiliary variables $\mathbf{w} = \{w_i\}_{i=1}^n$ indicating the component membership of the parameter $\boldsymbol{\theta}$, i.e., $\boldsymbol{\theta}_i = \boldsymbol{\tau}_{w_i}$. Instead of directly sampling $\boldsymbol{\theta}_i$, we sample $\{w_i\}_{i=1}^n$ and $\{\boldsymbol{\tau}_j\}_{j=1}^c$. The detailed sampling algorithm is given in Appendix II.

5.3 Prediction

After the burn-in iterations, we denote by $(\boldsymbol{\tau}^{(t)}, \mathbf{w}^{(t)})$ the parameters we sampled in the tth iteration. Given a new input $\mathbf{x}$, the predictive distribution of $\mathbf{y}$ is defined by

$$p(\mathbf{y}|\boldsymbol{\tau}^{(t)}, \mathbf{w}^{(t)}, \mathbf{x}) = \frac{\alpha \int p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}) G_0(d\boldsymbol{\theta}) + \sum_{j=1}^{c^{(t)}} n_j^{(t)} p(\mathbf{x}, \mathbf{y}|\boldsymbol{\tau}_j^{(t)})}{\alpha \int p(\mathbf{x}|\boldsymbol{\theta}) G_0(d\boldsymbol{\theta}) + \sum_{j=1}^{c^{(t)}} n_j^{(t)} p(\mathbf{x}|\boldsymbol{\tau}_j^{(t)})},$$

where

$$p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta}) = \int p(\mathbf{x}|\boldsymbol{\theta}, \boldsymbol{\eta}) p(\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{\eta}) p(\boldsymbol{\eta}) d\boldsymbol{\eta}.$$

Although $p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})$ can be shown to be a Gaussian, the parameterization is quite complicated. To make prediction more efficient, we use a different scheme. Note that

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\tau}^{(t)}, \mathbf{w}^{(t)}) = \int p(\mathbf{y}|\boldsymbol{\eta}, \boldsymbol{\theta}) p(\boldsymbol{\eta}|\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathbf{x}, \boldsymbol{\tau}^{(t)}, \mathbf{w}^{(t)}) d\boldsymbol{\eta} d\boldsymbol{\theta}.$$

In order to sample $\boldsymbol{\theta}^{(t)}$ from $p(\cdot|\mathbf{x}, \boldsymbol{\tau}^{(t)}, \mathbf{w}^{(t)})$, we first sample $\boldsymbol{\tau}_{new}^{(t)}$ from G_0 and then sample $\boldsymbol{\theta}^{(t)}$ based on

$$\left[\boldsymbol{\theta}^{(t)}|\mathbf{x}, \boldsymbol{\tau}^{(t)}, \mathbf{w}^{(t)}\right] \sim \sum_{j=1}^{c^{(t)}} n_j^{(t)} p(\mathbf{x}|\boldsymbol{\tau}_j) \delta_{\boldsymbol{\tau}_j^{(t)}} + \alpha \cdot p(\mathbf{x}|\boldsymbol{\tau}_{new}^{(t)}) \delta_{\boldsymbol{\tau}_{new}^{(t)}}.$$

Letting $\boldsymbol{\theta}^{(t)} = \{\mathbf{A}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)}, \mathbf{B}^{(t)}, \boldsymbol{\nu}^{(t)}, \boldsymbol{\Lambda}^{(t)}\}$, we sample the predicted response, denoted by $\mathbf{y}^{(t)}$, from the following distribution

$$p(\mathbf{y}^{(t)}|\boldsymbol{\theta}^{(t)}, \mathbf{x}) = \int p(\mathbf{y}^{(t)}|\boldsymbol{\eta}, \boldsymbol{\theta}^{(t)}) p(\boldsymbol{\eta}|\mathbf{x}, \boldsymbol{\theta}^{(t)}) d\boldsymbol{\eta}. \tag{6}$$

Here $p(\boldsymbol{\eta}|\mathbf{x}, \boldsymbol{\theta}^{(t)}) = N_r(\cdot|\mathbf{m}_\eta^{(t)}, \mathbf{V}_\eta^{(t)})$ where

$$\begin{aligned}
\mathbf{V}_\eta^{(t)} &= [(\mathbf{A}^{(t)})'(\boldsymbol{\Sigma}^{(t)})^{-1}\mathbf{A}^{(t)} + \mathbf{I}_r]^{-1}, \\
\mathbf{m}_\eta^{(t)} &= \mathbf{V}_\eta^{(t)}(\mathbf{A}^{(t)})'(\boldsymbol{\Sigma}^{(t)})^{-1}(\mathbf{x} - \boldsymbol{\mu}^{(t)}),
\end{aligned}$$

and the integral in equation (6) turns out to be $N_q(\mathbf{y}^{(t)}|\mathbf{m}_y^{(t)}, \mathbf{V}_y^{(t)})$, where

$$\begin{aligned}
\mathbf{V}_y^{(t)} &= \mathbf{B}^{(t)}\left[\mathbf{V}_\eta^{(t)} + \mathbf{m}_\eta^{(t)}(\mathbf{m}_\eta^{(t)})'\right](\mathbf{B}^{(t)})' + \boldsymbol{\Lambda}^{(t)}, \\
\mathbf{m}_y^{(t)} &= \mathbf{B}^{(t)}\mathbf{m}_\eta^{(t)} + \boldsymbol{\nu}^{(t)}.
\end{aligned}$$

This distribution generates the predicted response $\mathbf{y}^{(t)}$ for the tth iteration. Assume that posterior sampling is carried out for T time steps. Discarding the first T_0 iterations for the burn-in, the predicted $\mathbf{y}$ is

$$\mathbf{y} = \frac{1}{T - T_0} \sum_{t=T_0+1}^{T} \mathbf{y}^{(t)}.$$

6 Experimental Analysis

In this section we present the results of numerical experiments that evaluate the performance of our proposed Bayesian nonparametric models based on the matrix-variate Dirichlet process (MATDP) prior. We first present results for the DP-based multinomial probit regression (DP-MNP) presented in Section 4. We then discuss an experimental analysis of the matrix-variate DP Latent Factor Model (DP-LFM) in multivariate regression, multi-class classification, and multi-label prediction problems.

6.1 DP-MNP for Multi-class Classification

To evaluate the performance of our proposed DP-MNP method, we conducted empirical studies on several benchmark datasets and compared our method with two closely related classification methods: the multi-class Gaussian process classification method (GPC) (a degenerate sDP at $\alpha = \infty$), the model of Ibrahim and Kleinman (1998) and Xue et al. (2007) that we refer to as *DPC* (see Figure 1).

In the experiments we employed four multi-class classification datasets from the UCI database (`http://archive.ics.uci.edu/ml/`). These four datasets are the `Car Evaluation` database, the `Synthetic Control` database, the `Waveform` database, and the `Balance Scale` database, respectively.

- The `Car Evaluation` dataset was derived from a simple hierarchical decision model originally developed for the demonstration of DEX (Bohanec and Rajkovic 1990). It contains 1728 samples of 4 classes, each instance with 6 attributes.
- The `Synthetic Control` dataset was originally used for a clustering problem (Alcock and Manolopoulos 1999). It contains 600 samples of synthetically generated control charts, and each instance with 60 attributes. Moreover, there are six different classes of control charts, i.e., normal, cyclic, increasing trend, decreasing trend, upward shift, and downward shift. Here we treat this dataset as a six-class classification problem.
- The `Waveform` database contains 5000 samples of 3 classes of waves, and each instance with 21 attributes with continuous values between 0 and 6. In essence, each class is generated from a combination of 2 of 3 "base" waves.
- The `Balance Scale` database was originally generated to model psychological experimental results. It contain 625 samples with 3 classes, and each instance with 4 attributes. Specifically, each instance is classified as having the balance scale tip to the right, tip to the left, or be balanced, and the attributes are the left weight, the left distance, the right weight, and the right distance.

Table 1 gives a summary of these benchmark datasets. In our experiments, each dataset was randomly partitioned into two disjoint subsets as the training and test, with the percentage of the training data samples also given in Table 1. Ten random partitions were chosen for each dataset, and the average and standard deviation of their classification error rates over the test data were reported. For the sake of simplicity, in the following

experiments, the radial basis function (RBF) kernel, $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2)$, was employed and σ was set to the mean Euclidean distance among the input vectors. For our DP-MNP method, the other parameters were set as follows: $\alpha_a = 4$, $\alpha_b = 1$, $a_i = 4$ and $b_i = 1$ for $i = 1, \ldots, m+1$. Additionally, we set $\rho = q$ and $\mathbf{R}_{11} = \mathbf{I}_{q-1} + \frac{1}{q-1}\mathbf{1}_{q-1}\mathbf{1}'_{q-1}$ (the equicorrelation matrix). These simple settings were found to be effective, although we make no claims of optimality.

Dataset	q	p	k	n/k
`Car Evaluation`	4	6	1728	10%
`Synthetic Control`	6	60	600	40%
`Waveform`	3	21	5000	2.5%
`Balance Scale`	3	4	625	40%

Table 1: Summary of the benchmark datasets: q—the number of classes; p—the dimensionality of the input vector; n—the number of training samples; k—the size of the dataset.

Table 2 shows the corresponding test results. From the table, we can see that the overall performance of our DP-MNP method is slightly better than the two competing methods. In the comparison to GPC, the difference is presumably due to the ability of the DP-MNP to capture relationships among data points, whereas in the comparison to the DPC the DP-MNP profits from its ability to exploit relationships among the components of the response vector.

Note that the dimensionality of the response q is not large in the four datasets. For DPC, the clustering property of DP could place all of the regression vectors in a single class, enforcing too much sharing. Thus, we took a larger mass parameter in implementing DPC, as suggested by Bush et al. (2010). Recall that the limiting case of the DPC model at $\alpha = \infty$ is identical to a degenerate DP-MNP method at $\alpha = 0$, while the GPC method can be regarded as a degenerate sDP model at $\alpha = \infty$.

6.2 DP-LFM for Multivariate Regression

We test the effectiveness of our proposed nonparametric factor analyzers in a collection of experiments. We first demonstrate our DP-LFM in the multivariate regression setting using the `chemometrics` dataset and the `robot arm` dataset. The `chemometrics` data taken from Skagerberg et al. (1992) were used in Breiman and Friedman (1997) to

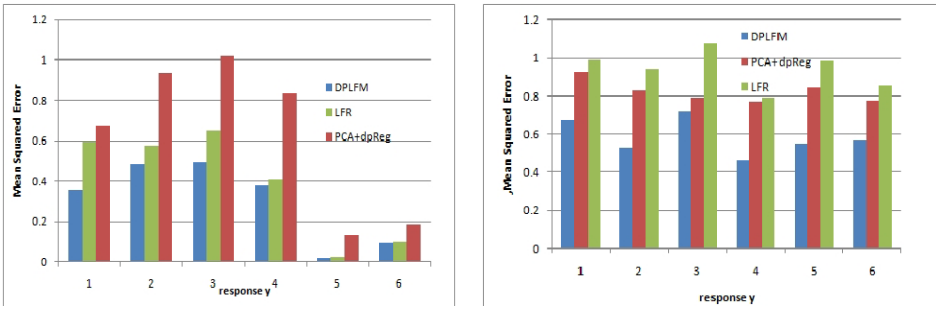
Dataset	GPC	DPC	DP-MNP
`Car Evaluation`	26.64 ($\pm$1.24)	28.18 ($\pm$1.51)	**26.31** ($\pm$1.28)
`Synthetic Control`	16.32 ($\pm$0.98)	16.42 ($\pm$1.97)	**15.42** ($\pm$1.34)
`Waveform`	17.96 ($\pm$0.12)	16.50 ($\pm$1.14)	**15.84** ($\pm$0.85)
`Balance Scale`	16.22 ($\pm$2.10)	16.31 ($\pm$2.63)	**15.03** ($\pm$1.93)

Table 2: Classification error rates (%) and standard deviations on the four datasets.

analyze their regression methods. The `robot arm` dataset was used by Teh et al. (2005) for modeling in the domain of multi-joint robot arm dynamics. Both datasets have six responses. The `chemometrics` data has 58 samples and the dimensionality of $\mathbf{x}$ is 22. The `robot arm` data has 1500 samples and the dimensionality of $\mathbf{x}$ is 12.

In the experiments we preprocessed the data to have zero mean and unit variance. We used the same setup for the hyperparameters in the two datasets: $a_\alpha = -2$, $b_\alpha = 3$, $a_\mu = a_\nu = 1$, $b_\mu = b_\nu = 1$, $a_\sigma = a_\lambda = 1$ and $b_\sigma = b_\lambda = 1$. Our competitor is the Dirichlet process regression model (dpReg) in Shahbaba and Neal (2009). We tested dpReg's performance on the data preprocessed by principal component analysis.

For both the datasets, we set the latent variable dimensionality equal to four for the DP-LFM. For comparison, we also projected the data onto a four-dimensional subspace and fit a dpReg model in that subspace. Furthermore, to evaluate the benefits of a nonlinear model, we compared to the West (2003) model (LFR for short) by setting α in DP-LFM to zero. We used 35 data samples in the `chemometrics` and 1000 data samples in the `robot arm` for training respectively. We compared the mean squared error on each response and summarize the results in Figure 3(a) and Figure 3(b). Different bar groups correspond to different regression responses. The experimental results demonstrate that the DP-LFM outperforms PCA+dpReg and LFR, illustrating the advantages that accrue to a model that can capture nonlinearity and can perform supervised dimensionality reduction.



(a) Chemometric Data (b) Robot Arm Data

Figure 3: Performance comparison of DP-LFM, dpReg, and LFR on datasets for regression.

6.3 DP-LFM for Multi-Class Classification on Synthetic Data

In this experiment, we focused on a four-way classification problem on synthetic data similar to Shahbaba and Neal (2009), but in a slightly different generative setting. Setting the dimensions of $\mathbf{x}$ and $\boldsymbol{\eta}$ to be 10 and 2 respectively, we first generated two

components and related parameters $\{\boldsymbol{\theta}_i = \{\mathbf{A}_i, \mathbf{B}_i, \boldsymbol{\mu}_i, \boldsymbol{\nu}_i, \boldsymbol{\Sigma}_i, \boldsymbol{\Lambda}_i\}\}_{i=1}^2$ as follows

$$\begin{aligned} \mathbf{A}_i(j,k) &\sim N(\cdot|0, \sigma_{A_i}^2), \\ \boldsymbol{\mu}_i(j) &\sim N(\cdot|0, 2^2), \\ \log(\boldsymbol{\Sigma}_i(j,j)^2) &\sim N(\cdot|0, 1), \\ \log(\sigma_{A_i}^2) &\sim N(\cdot|0, 2^2). \end{aligned}$$

The parameters for generating $\mathbf{y}$ were specified similarly. Afterwards, we generated the latent variable $\boldsymbol{\eta}$ from $N_2(\boldsymbol{\eta}|\mathbf{0}, \mathbf{I}_2)$ and then randomly chose the component $\boldsymbol{\eta}$ belonging to, say $\boldsymbol{\theta}_i$. Then $\mathbf{x}$ and $\mathbf{y}$ were sampled from a Gaussian distribution and a multinomial logit distribution respectively.

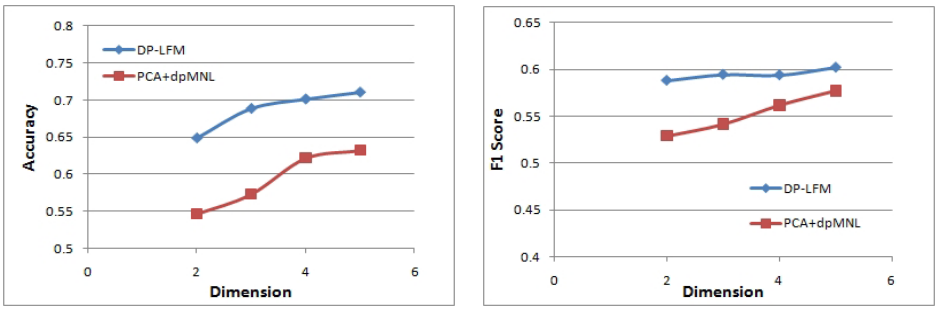
The goal in this experiment was to evaluate the advantage of doing dimensionality reduction and classification jointly. Evaluation is based on the F1-score (Murphy 2012). Let $\hat{y}_i \in \{0,1\}$ be the predicted label, and $y_i \in \{0,1\}$ be the true label. Then the "accuracy" is defined as $A \triangleq \frac{\sum_i y_i\hat{y}_i + (1-y_i)(1-\hat{y}_i)}{\sum_i \hat{y}_i + (1-\hat{y}_i)}$, the "precision" as $P \triangleq \frac{\sum_i y_i\hat{y}_i}{\sum_i \hat{y}_i}$ and the "recall" as $R \triangleq \frac{\sum_i y_i\hat{y}_i}{\sum_i y_i}$. Accordingly, the "F1-score" is $F_1 \triangleq \frac{2PR}{R+P} = \frac{2\sum_i y_i\hat{y}_i}{\sum_i (y_i+\hat{y}_i)}$. In the multi-class ($q$-class) case, there are two approaches to generalize the F1-score: "macro-averaged F1" and "micro-averaged F1." Let $(\hat{y}_{ij}, \ldots, \hat{y}_{iq})^T \in \{0,1\}^q$ and $(y_{ij}, \ldots, y_{iq})^T \in \{0,1\}^q$ be the predicted and true label vector, respectively. Then the F1-Macro is defined as $\frac{1}{q}\sum_{j=1}^q \frac{2\sum_i y_{ij}\hat{y}_{ij}}{\sum_i (y_{ij}+\hat{y}_{ij})}$, while the F1-Micro is defined as $\frac{2\sum_{j=1}^q \sum_i y_{ij}\hat{y}_{ij}}{\sum_{j=1}^q \sum_i (y_{ij}+\hat{y}_{ij})}$.

We compared two models: the first was our DP-LFM with the original $\mathbf{x}$ as its input, and the second was the dpMNL model (Shahbaba and Neal 2009) with the input preprocessed by principal component analysis (PCA). More specifically, for the DP-LFM model, we first set the dimensionality of the latent variable to be d and trained it with the original data. For the dpMNL, we first projected the original data into a d-dimensional space using PCA and trained the dpMNL model on the transformed data.

The hyperparameters for the matrix-variate prior were set as follows: $\boldsymbol{\Sigma}_\Phi^{\text{MDP}} = \boldsymbol{\Sigma}_\Psi^{\text{MDP}} = \mathbf{I}_r$, $a_\nu^{\text{MDP}} = 10$, and $b_\nu^{\text{MDP}} = 1$. Note that for simplicity we directly set $\boldsymbol{\Sigma}_\Phi^{\text{MDP}}$ and Σ_Ψ^{MDP} to be identity matrices and eliminated their hyperparameters ρ^{MDP} and $\mathbf{R}^{\text{MDP}}$ as in the previous subsection. The hyperparameters a_α and b_α were set to -2 and 3 respectively. All the other hyperparameters were set to be 1 in this experiment.

We randomly generated 20 datasets, each of which contained 100 data points for training and 1900 for test. We ran 5000 MCMC iterations for each model and used the last 4000 iterations for prediction. We adopted accuracy and F1-MACRO (the average F1-Score over all categories) as performance metrics and evaluated the two models' average performance on these datasets. The performance of the two models was compared for different choices of d ranging from two to five and is depicted in Figure 4. From the figure, we can see that handling dimensionality reduction and classification jointly improves the performance. It should be noted that the experiment does not establish that DP-LFM is a better model than dpMNL. Indeed, PCA could be an inappropriate preprocessor that leads to dpMNL's poorer performance. The experiment only demon-

strates that for high-dimensional classification problem, it may be a bad idea to separate dimensionality reduction and classification.



(a) Accuracy (b) F1 Score

Figure 4: Performance comparison of DP-LFM and dpMNL on synthetic datasets.

6.4 Application to Parkinson's disease data

In this section, we test our DP-LFM on real world datasets for classification. We used the `Parkinson` dataset, which was obtained from the UCI Repository. Each instance has 22 features and a binary label indicating whether he/she is a patient with Parkinson's disease. Note that Shahbaba and Neal (2009) used PCA to preprocess the data and chose the first ten principal components in implementing their dpMNL model. Our method does not need this preprocessor, because it implements classification and dimensionality reduction simultaneously. To compare our DP-LFM with dpMNL, we correspondingly set the number of factor loadings r in our model equal to ten.

The hyperparameters for the matrix variate DP prior were set as in the above section. The other hyperparameters were set as follows: $a_\alpha = -2$, $b_\alpha = 3$, $a_\mu = a_\nu = 1$, $b_\mu = b_\nu = 2$, $a_\sigma = a_\lambda = 1$ and $b_\sigma = b_\lambda = 2$. In this experiment, we ran MCMC for 5000 iterations, discarding the first 1000 burn-in iterations. The typical number of components in the MCMC iterations ranges from 5 to 7. We compared DP-LFM with baselines provided in Shahbaba and Neal (2009), which are summarized as follows:

- MNL & qMNL: Multinomial logit models and MNL with quadratic terms.
- SVM & SVM-RBF: Support vector machines and SVM with RBF-Kernel.
- dpMNL: Dirichlet process multinomial logit model (Shahbaba and Neal 2009).

As in Shahbaba and Neal (2009), we used a five-fold cross validation scheme to get a reliable performance estimate of our proposed models. The evaluation metrics we used to compare algorithms were "accuracy" and F1-Score. The results are reported in Table 3, which shows the average performance and standard deviation for five randomly split datasets. It can be seen that DP-LFM outperforms the other models. We attribute the performance improvement to two reasons. First, our dimensionality reduction procedure

is supervised. While dpMNL used PCA to preprocess the data for their model, our model does dimensionality reduction and classification jointly and estimates the factor loading matrix $\mathbf{A}$ with the information from the given labels. Second, our dimensionality reduction was carried out locally instead of globally. While PCA globally reduces the data to a ten-dimensional subspace, our model assumes that each instance has its own factor loading matrix. By imposing a MATDP prior, we cluster data into regions and estimate the region-specific factor loading matrix. This also provides potential evidence that DP-LFM and dpMNL outperform the other discriminative methods. That is, the clustering property of the DP can make DP-LFM and dpMNL reveal some information about the underlying structure in the data (Shahbaba and Neal 2009).

models	**Performance**	
	Accuracy	F1
MNL	0.856 ($\pm$0.022)	0.797 ($\pm$0.028)
qMNL	0.861 ($\pm$0.015)	0.797 ($\pm$0.021)
SVM	0.872 ($\pm$0.023)	0.806 ($\pm$0.028)
SVM-RBF	0.872 ($\pm$0.027)	0.799 ($\pm$0.032)
dpMNL	0.877 ($\pm$0.033)	0.826 ($\pm$0.025)
DP-LFM	**0.882** ($\pm$0.035)	**0.843** ($\pm$0.024)

Table 3: Performance comparison for Parkinson's disease data.

6.5 DP-LFM for Multi-Label Prediction

Finally, we evaluated our DP-LFM model in the setting of the multi-label prediction problem, where each input vector $\mathbf{x}_i$ is associated with a vector of responses $\mathbf{y}_i$. The datasets we used here were `Yeast` and `Scene` from the UCI repository. The `Yeast` dataset consists of gene-expression data. The number of data samples is 2417, with 1500 data samples for training and the others for test. Each input vector has 103 features and may belong to any of the 14 predefined groups. The `Scene` dataset has a total of 2407 data samples, 1211 for training and 1196 for test. The feature dimensionality in this dataset is 294 and the number of classes for each data instance is 6.

For both datasets, we preprocessed the data so that the input vectors for our algorithm had zero mean and unity variance. The hyperparameters for the matrix-variate prior were defined identically to those in the above experiment. The number of factor loadings r was chosen to be 20. Hyperparameters for our model were set as follows: $a_\alpha = -2$, $b_\alpha = 3$, $a_\mu = a_\nu = 5$, $b_\mu = b_\nu = 2$, $a_\sigma = a_\lambda = 1$ and $b_\sigma = b_\lambda = 2$.

In this experiment, we ran MCMC for 3000 iterations, discarding the first 1000 burn-in iterations. The typical number of components in the MCMC iterations ranged from 4 to 6 for the `Yeast` dataset, and from 18 to 20 for the `Scene` dataset. We compared our DP-LFM with the following algorithms:

- PCA: Principal component analysis which projects the data into a latent subspace in an unsupervised manner. A nearest-neighbor classifier is trained for the data

after projection.

- PLS: Partial least squares which finds the common structure between explanatory variables and responses.
- SPPCA and SSPPCA: Supervised versions of probabilistic principal component analysis. SPPCA incorporates the label in the training data to guide the projection, while SSPPCA further leverages the information of the explanatory variables in the test data.
- SInf-CCA: Semi-supervised infinite version of canonical correlation analysis which aims to principally find the correlation between the exploratory variables and responses.

These algorithms have been studied by Yu et al. (2006) and Rai and Daumé III (2009). Note that the SInf-CCA of Rai and Daumé III (2009) is a Bayesian nonparametric method. We did not implement the dpMNL model (Shahbaba and Neal 2009) given that it was designed for classification and not for multi-label prediction. Moreover, the data samples in the two datasets we studied are both high-dimensional, which presents a scalability issue for the dpMNL.

We summarize the performance of our algorithm and the other algorithms in Table 4. The evaluation metrics are F1-MACRO and F1-MICRO (the average F1-measure over all data samples) (Yu et al. 2006). The results of the compared algorithms are directly cited from Yu et al. (2006) and Rai and Daumé III (2009), where we have chosen the best-performing algorithms from their comparison. It can be seen from the table that our algorithm outperforms the other ones in three out of the four entries. Moreover, while SSPPCA outperforms our algorithm in the F1-MACRO metric on the `Yeast` dataset, it must be kept in mind that SSPPCA uses test data during training, which may not be feasible in real-world applications.

models	Yeast		Scene	
	F1-MACRO	F1-MICRO	F1-MACRO	F1-MICRO
PCA	0.3723	0.5537	0.2857	0.2834
PLS	0.3799	0.5208	0.5745	0.5831
SPPCA	0.3859	0.5571	0.5173	0.5309
SSPPCA	**0.3976**	0.6012	0.5537	0.5783
Inf-CCA	0.3463	0.4954	0.3742	0.3825
DP-LFM	0.3903	**0.6389**	**0.5871**	**0.6014**

Table 4: Performance comparison for multi-label prediction.

7 Conclusion and Future Work

We have proposed the notion of matrix-variate DP priors. Based on this notion, we have developed a Bayesian nonparametric discriminative model and a Bayesian non-

parametric latent factor model for multivariate supervised learning problems. We have also devised MCMC algorithms for inference and prediction. Our models are nonlinear. Moreover, our nonparametric latent factor model has the advantage of performing dimensionality reduction and regression or classification jointly.

In our nonparametric latent factor model, the dimensionality of the latent vector (i.e., the number of factors) is assumed to be prespecified by practitioners. It is desirable to address the automatic choice of this value. A potential approach for handling this issue is to assign a sparsity prior for loading matrices as in West (2003) and Carvalho et al. (2010). Another possibility is to make use of nonparametric model-averaging methods, in particular methods based on Beta process priors (Paisley and Carin 2009; Rai and Daumé III 2009).

References

Albert, J. H. and Chib, S. (1993). "Bayesian Analysis of Binary and Polychotomous Response Data." *Journal of the American Statistical Association*, 88(422): 669–679. 263

Alcock, R. J. and Manolopoulos, Y. (1999). "Time-Series Similarity Queries Employing a Feature-Based Approach." In *Seventh Hellenic Conference on Informatics*. Ioannina, Greece. 272

Antoniak, C. E. (1974). "Mixtures of Dirichlet Processes with applications to Bayesian nonparametric problems." *The Annals of Statistics*, 2: 1152–1174. 260

Blackwell, D. and MacQueen, J. B. (1973). "Ferguson distributions via Pólya urn schemes." *The Annals of Statistics*, 1: 353–355. 262

Bohanec, M. and Rajkovic, V. (1990). "Expert system for decision making." *Sistemica*, 1(1): 145–157. 272

Breiman, L. and Friedman, J. (1997). "Predicting multivariate responses in multiple linear regression (with discussion)." *Journal of the Royal Statistical Society, B*, 59(1): 3–54. 273

Bush, C. A., Lee, J., and MacEachern, S. N. (2010). "Minimally informative prior distributions for non-parametric Bayesian analysis." *Journal of the Royal Statistical Society, B*, 72(2): 253–268. 266, 273

Bush, C. A. and MacEachern, S. N. (1996). "A semiparametric Bayesian model for randomised block designs." *Biometrika*, 83: 275–285. 260, 283

Caruana, R. (1997). "Multitask Learning." *Machine Learning*, 28(1): 41–75. 259

Carvalho, C. M., Chang, J., Lucas, J. E., Nevins, J. R., Wang, Q., and West, M. (2010). "High-Dimensional Sparse Factor Modeling: Applications in Gene Expression Genomics." *Journal of the American Statistical Association*, 103(484): 1438–1456. 268, 269, 279

Chen, M., Silva, J., Paisley, J., Wang, C., Dunson, D., and Carin, L. (2010). "Compressive Sensing on Manifolds Using a Nonparametric Mixture of Factor Analyzers: Algorithm and Performance Bounds." Technical report, Electrical & Computer Engineering Department, Duke University, USA. 269

Denison, D. G. T., Holmes, C. C., Mallick, B. K., and Smith, A. F. M. (2002). *Bayesian Methods for Nonlinear Classification and Regression*. New York: John Wiley and Sons. 263

Dunson, D. B., Pillai, N., and Park, J.-H. (2007). "Bayesian Density Regression." *Journal of the Royal Statistical Society Series B*, 69(2): 163–183. 259, 267

Dunson, D. B., Xue, Y., and Carin, L. (2008). "The Matrix Stick-Breaking Process." *Journal of the American Statistical Association*, 103(481): 317–327. 267

Escobar, M. D. and West, M. (1995). "Bayesian Density Estimation and Inference Using Mixtures." *Journal of the American Statistical Association*, 90: 577–588. 260, 283

Ferguson, T. S. (1973). "A Bayesian Analysis of Some Nonparametric Problems." *The Annals of Statistics*, 1: 209–230. 260

Gelfand, A. E., Kottas, A., and MacEachern, S. N. (2005). "Bayesian Nonparametric Spatial Modeling with Dirichlet Process Mixing." *Journal of the American Statistical Association*, 100: 1021–1035. 259, 260, 267

Golub, G. H. and Loan, C. F. V. (1996). *Matrix Computations*. Baltimore, MD: The Johns Hopkins University Press. 265

Görür, D. and Rasmussen, C. E. (2007). "Dirichlet Process Mixtures of Factor Analysers." In *Fifth Workshop on Bayesian Inference in Stochastic Processes*. 269

Griffiths, T. L. and Ghahramani, Z. (2005). "Infinite latent feature models and the Indian buffet process." In *Advances in Neural Information Processing Systems*. 260

Gupta, A. K. and Nagar, D. K. (2000). *Matrix Variate Distributions*. Chapman & Hall/CRC. 261

Hannah, L. A., Blei, D. M., and Powell, W. B. (2010). "Dirichlet Process Mixtures of Generalized Linear Models." In *The Thirteenth International Conference on AI and Statistics*. 260

Holmes, C. C. and Held, L. (2006). "Bayesian auxiliary variable methods for binary and multinomial regression." *Bayesian Analysis*, 1(1): 145–168. 263

Ibrahim, J. G. and Kleinman, K. P. (1998). "Semiparametric Bayesian Methods for Random Effects Models." In Dey, D., Müller, P., and Sinha, D. (eds.), *Practical Nonparametric and Semiparametric Bayesian Statistics*, 89–114. New York: Springer-Verlag. 259, 266, 272

MacEachern, S. N. (1998). "Computational Methods for Mixture of Dirichlet Process Models." In Dey, D., Müller, P., and Sinha, D. (eds.), *Practical Nonparametric and Semiparametric Bayesian Statistics*, 23–43. New York: Springer-Verlag. 260

— (1999). "Dependent Nonparametric Processes." In *The Section on Bayesian Statistical Science*, 50–55. American Statistical Association. 259

MacEachern, S. N. (2000). "Dependent Dirichlet processes." Technical report, Ohio State University, Department of Statistics. 260, 262

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979). *Multivariate Analysis*. New York: Academic Press. 264, 283

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. Cambridge, Massachusetts: The MIT Press. 275

Neal, R. M. (2000). "Markov chain sampling methods for Dirichlet process mixture models." *Journal of Computational and Graphical Statistics*, 9: 249–265. 260, 284

— (2003). "Slice sampling." *The Annals of Statistics*, 31(3): 705–767. 284

Ng, A. Y. and Jordan, M. I. (2002). "On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes." In *Advances in Neural Information Processing Systems 14*. 261

Paisley, J. and Carin, L. (2009). "Nonparametric factor analysis with beta process priors." In *Proceedings of the 26th Annual International Conference on Machine Learning*. 279

Rai, P. and Daumé III, H. (2009). "Multi-Label Prediction via Sparse Infinite CCA." In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 22*, 1518–1526. 260, 278, 279

— (2010). "Infinite Predictor Subspace Models for Multitask Learning." In *Proceedings of the Conference on Artificial Intelligence and Statistics (AISTATS)*. Sardinia, Italy. 260

Schölkopf, B. and Smola, A. (2002). *Learning with Kernels*. The MIT Press. 263

Shahbaba, B. and Neal, R. (2009). "Nonlinear Models Using Dirichlet Process Mixtures." *Journal of Machine Learning Research*, 10(2): 1829–1850. 260, 269, 274, 275, 276, 277, 278, 284

Skagerberg, B., MacGregor, J., and Kiparissides, C. (1992). "Multivariate data analysis applied to low-density polythylene reactors." *Chemometrics and intelligent laboratory systems*, 14: 341–356. 273

Teh, Y. W., Seeger, M., and Jordan, M. I. (2005). "Semiparametric Latent Factor Models." In *Workshop on AI and Statistics 10*. 274

Tewari, A. and Bartlett, P. L. (2007). "On the consistency of multiclass classification methods." *Journal of Machine Learning Research*, 8: 1007–1025. 259

West, M. (2003). "Bayesian factor regression models in the "large p, small n" paradigm." In Bernardo, J. M., Bayarri, M. J., Berger, J. O., Dawid, A. P., Heckerman, D., Smith, A. F. M., and West, M. (eds.), *Bayesian Statistics 7*, 723–732. Oxford University Press. 260, 268, 269, 274, 279

Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. (2007). "Multi-Task learning for classification with Dirichlet Process priors." *Journal of Machine Learning Research*, 8: 35–63. 259, 266, 272

Yu, S., Yu, K., Tresp, V., Kriegel, H.-P., and Wu, M. (2006). "Supervised probabilistic principal component analysis." In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 464–473. New York, NY, USA. 278

Zhang, Z., Dai, G., and Jordan, M. I. (2010). "Matrix-variate Dirichlet process mixture models." In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*. 262

Appendix I: The MCMC Algorithm for Multivariate Regression

In this paper we define $g_j(\mathbf{x}) = K(\mathbf{x}_j, \mathbf{x})$ for $j = 1, \ldots, n$. This implies that we have n basis functions; that is, $m = n$. We can use Gibbs sampling to draw $[\mathbf{B}_{i\,i=1}^{\,n}, \boldsymbol{\Sigma}, \boldsymbol{\Lambda}, \alpha|\mathbf{Z}]$. The required full conditionals are

(a) $[(\mathbf{B}_i, w_i)|(\mathbf{B}_{-i}, \mathbf{w}_{-i}), \alpha, \boldsymbol{\Lambda}, \boldsymbol{\Sigma}, \mathbf{Z}]$ for $i = 1, \ldots, n$;

(b) $[\mathbf{Q}_k|\mathbf{w}, \boldsymbol{\Lambda}, \boldsymbol{\Sigma}, \alpha, \mathbf{Z}]$ for $k = 1, \ldots, c$;

(c) $[\boldsymbol{\Sigma}^{-1}|\mathbf{Z}, \mathbf{B}, \mathbf{R}, \rho]$;

(d) $[\lambda_i^{-1}|\{\boldsymbol{\beta}_i^{(k)}\}_{k=1}^{c}, \boldsymbol{\Sigma}, a_i, b_i]$ for $i = 1, \ldots, n{+}1$;

(e) $[\alpha|a_\alpha, b_\alpha, c]$.

The Gibbs sampler exploits the simple structure of the conditional posterior for each $\mathbf{B}_i$. That is, for $i = 1, \ldots, n$, the conditional distribution is given by

$$[\mathbf{B}_i|\mathbf{B}_{-i},\ \mathbf{Z}, \boldsymbol{\Lambda}, \boldsymbol{\Sigma}] \propto q_0 N_q(\mathbf{z}_i|\mathbf{B}_i'\mathbf{g}_i,\ \boldsymbol{\Sigma}) N_{n+1}(\mathbf{B}_i|\mathbf{0},\ \boldsymbol{\Lambda}\otimes\boldsymbol{\Sigma}) + \sum_{j\neq i} q_j \delta_{\mathbf{B}_j}, \tag{7}$$

where $q_j = N_q(\mathbf{z}_i|\mathbf{B}_j'\mathbf{g}_i,\ \mathbf{\Sigma})$ and

$$\begin{aligned} q_0 &= \alpha \int N_q(\mathbf{z}_i|\mathbf{B}_i'\mathbf{g}_i,\ \mathbf{\Sigma}) N_{n+1,q}(\mathbf{B}_i|\mathbf{0},\ \mathbf{\Lambda}\otimes\mathbf{\Sigma}) d\mathbf{B}_i \\ &= \alpha \cdot N_q(\mathbf{z}_i|\mathbf{0},\ (\mathbf{g}_i'\mathbf{\Lambda}\mathbf{g}_i + 1)\mathbf{\Sigma}). \end{aligned}$$

Note that $N_q(\mathbf{z}_i|\mathbf{B}_j'\mathbf{g}_i,\ \mathbf{\Sigma})$ and $N_q(\mathbf{z}_i|\mathbf{0},\ (\mathbf{g}_i'\mathbf{\Lambda}\mathbf{g}_i+1)\mathbf{\Sigma})$ are multivariate singular normal distributions, which can be computed via the nonzero eigenvalues of $\mathbf{\Sigma}$ (Mardia et al. 1979). According to equation (2), (7) thus reduces to

$$[\mathbf{B}_i|\mathbf{B}_{-i}, \mathbf{z}_i, \mathbf{\Lambda}, \mathbf{\Sigma}] \propto\ q_0 \cdot N_{n+1,q}(\mathbf{B}_i|\mathbf{A}_i\mathbf{g}_i\mathbf{z}_i',\ \mathbf{A}_i \otimes \mathbf{\Sigma}) + \sum_{k=1}^{c} n_{k(-i)} q_k \delta_{\mathbf{Q}_k},$$

where $\mathbf{A}_i = (\mathbf{\Lambda}^{-1} + \mathbf{g}_i\mathbf{g}_i')^{-1}$. Thus, given $\mathbf{B}_{-i}$, with probability proportional to $n_{k(-i)}q_k$, we draw $\mathbf{B}_i$ from distribution $\delta_{\mathbf{Q}_k}$, or with probability proportional to q_0, we draw $\mathbf{B}_i$ from $N_{n+1,n}(\cdot|\mathbf{A}_i\mathbf{g}_i\mathbf{z}_i',\ \mathbf{A}_i \otimes \mathbf{\Sigma})$. Here we again use the Sherman-Morrison-Woodbury formula to calculate $\mathbf{A}_i$. That is,

$$\mathbf{A}_i = (\mathbf{\Lambda}^{-1} + \mathbf{g}_i\mathbf{g}_i')^{-1} = \mathbf{\Lambda} - \mathbf{\Lambda}\mathbf{g}_i(1 + \mathbf{g}_i'\mathbf{\Lambda}\mathbf{g}_i)^{-1}\mathbf{g}_i'\mathbf{\Lambda},$$

which involves reciprocal computations.

To speed mixing of the Markov chain, Bush and MacEachern (1996) suggested resampling the $\mathbf{Q}_k$ after every step. For each $k = 1, \ldots, c$, we have

$$[\mathbf{Q}_k|\mathbf{Z}, \mathbf{w}, \mathbf{\Lambda}, \mathbf{\Sigma}] \propto N_{n+1,q}(\mathbf{Q}_k|\mathbf{0},\ \mathbf{\Lambda}\otimes\mathbf{\Sigma}) \prod_{i:\ w_i=k} N_q\Big(\mathbf{z}_i|\mathbf{Q}_k'\mathbf{g}_i,\ \mathbf{\Sigma}\Big),$$

from which it follows that the conditional distribution of $\mathbf{Q}_k$ is given by equation (3).

The update of $\mathbf{\Sigma}_{11}$ is given by

$$[\mathbf{\Sigma}_{11}^{-1}|\mathbf{Z}, \mathbf{B}, \rho, \mathbf{R}_{11}] \sim W_{q-1}\left(\mathbf{\Sigma}_{11}^{-1}\middle|\rho + n, \mathbf{R}_{11} + \sum_{i=1}^{n} \mathbf{H}_q(\mathbf{z}_i - \mathbf{B}_i'\mathbf{g}_i)(\mathbf{z}_i - \mathbf{B}_i'\mathbf{g}_i)'\mathbf{H}_q'\right).$$

Since the λ_i for $i = 1, 2, \ldots, n+1$ are only dependent on the $\mathbf{Q}_k$, we use the Gibbs sampler to update them from their own conditional distributions as

$$[{\lambda_i}^{-1}|\mathbf{Q}, a_i, b_i] \sim\ \mathrm{Ga}\left({\lambda_i}^{-1}\middle|\frac{a_i + qc}{2}, \frac{b_i + \sum_{k=1}^{c} (\boldsymbol{\beta}_i^{(k)})'\mathbf{H}_q\mathbf{\Sigma}_{11}^{-1}\mathbf{H}_q'\boldsymbol{\beta}_i^{(k)}}{2}\right),$$

where $(\boldsymbol{\beta}_i^{(k)})'$ is the ith row of $\mathbf{Q}_k$.

As for the estimate of α, we directly follow the data augmentation technique proposed by Escobar and West (1995). That is, given the currently sampled values of c and α, one samples a random variable ω from Beta distribution $\mathrm{Be}(\alpha+1, n)$; one then samples a new α from the following mixture as

$$[\alpha|\omega, c] \sim \pi_0 \mathrm{Ga}(\alpha|a_\alpha + c, b_\alpha - \log(\omega)) + (1 - \pi_0)\mathrm{Ga}(\alpha|a_\alpha + c - 1, b_\alpha - \log(\omega))$$

with $\pi_0 = \frac{\alpha+c-1}{a_\alpha+c-1+n(b_\alpha-\log(\omega))}$.

Appendix II: Posterior Sampling for DP-LFM

Given the discreteness of the random measure G, we assume that there are c distinct values among $\{\boldsymbol{\theta}_i\}_{i=1}^n$, denoted $\{\boldsymbol{\tau}_j = (\mathbf{A}_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \mathbf{B}_j, \boldsymbol{\nu}_j, \boldsymbol{\Lambda}_j)\}_{j=1}^c$. We further introduce n auxiliary variables $\mathbf{w} = \{w_i\}_{i=1}^n$ indicating the component membership of the parameter $\boldsymbol{\theta}$, i.e., $\boldsymbol{\theta}_i = \boldsymbol{\tau}_{w_i}$. Instead of directly sampling $\boldsymbol{\theta}_i$, we sample $\{w_i\}_{i=1}^n$ and $\{\boldsymbol{\tau}_j\}_{j=1}^c$. The main sampling algorithm is listed as follows:

- Let $(\mathbf{x}_i, \mathbf{y}_i)$ be the last observed data instance, and denote by $n_{j(-i)}$ the number of samples except $(\mathbf{x}_i, \mathbf{y}_i)$ in the jth component. We have the following posterior distribution for w_i:

$$p(w_i|(\mathbf{x}_i, \mathbf{y}_i), \boldsymbol{\tau}, \alpha, G_0) \propto \begin{cases} n_{j(-i)} p((\mathbf{x}_i, \mathbf{y}_i)|\boldsymbol{\eta}_i, \boldsymbol{\tau}_j) & w_i = j \leq c, \\ \alpha \int p((\mathbf{x}_i, \mathbf{y}_i)|\boldsymbol{\eta}_i, \boldsymbol{\theta}) G_0(d\boldsymbol{\theta}) & w_i = c+1. \end{cases}$$

  Here the likelihood of $(\mathbf{x}_i, \mathbf{y}_i)$ is

$$p((\mathbf{x}_i, \mathbf{y}_i)|\boldsymbol{\tau}_j, \boldsymbol{\eta}_i) = N_q(\mathbf{y}_i|\mathbf{B}_j\boldsymbol{\eta}_i + \boldsymbol{\nu}_j,\ \boldsymbol{\Lambda}_j) N_p(\mathbf{x}_i|\mathbf{A}_j\boldsymbol{\eta}_i + \boldsymbol{\mu}_j,\ \boldsymbol{\Sigma}_j).$$

  For simplicity, we do not write the likelihood function explicitly in the rest of the paper.

  In many cases, the integral in the above sampling formula can be difficult to evaluate. To circumvent this issue, we avail ourselves of a trick proposed by Neal (2000); Shahbaba and Neal (2009) where we first sample m additional components $\{\boldsymbol{\tau}_{c+1}, \boldsymbol{\tau}_{c+2}, \ldots, \boldsymbol{\tau}_{c+m}\}$ independently from G_0 and then sample the component which $\{\mathbf{x}_i, \mathbf{y}_i\}$ belongs to. If $\{\mathbf{x}_i, \mathbf{y}_i\}$ belongs to the component in $\{\boldsymbol{\tau}_{c+1}, \tau_{c+2}, \ldots, \tau_{c+m}\}$, a new component is generated and we set w_i to be $c+1$.

- Denoting by n_j the number of $(\mathbf{x}_i, \mathbf{y}_i)$ with $w_i = j$, we sample $\boldsymbol{\mu}_j$ according to the $N_p(\cdot|\mathbf{m}_j, \mathbf{V}_j)$, where

$$\mathbf{V}_j = (\boldsymbol{\Sigma}_j^{-1} + \mathrm{diag}(\mathbf{v}_\mu)^{-2})^{-1}$$
$$\mathbf{m}_j = \mathbf{V}_j \left[\boldsymbol{\Sigma}_j^{-1} (\sum_{w_i = j} \mathbf{x}_i - \mathbf{A}_j\boldsymbol{\eta}_i) + \mathrm{diag}(\mathbf{v}_\mu)^{-2} \mathbf{m}_\mu \right].$$

- The posterior distribution of $\boldsymbol{\Sigma}_j = \mathrm{diag}(\sigma_{j1}^2, \ldots, \sigma_{jp}^2)$ does not have a closed form. However, we can use slice sampling (Neal 2003) to sample from the following distribution:

$$p\Big(\log(\sigma_{jl}^2)\Big|\{\mathbf{x}_i\}_{i=1}^n, \boldsymbol{\mu}_j, \mathbf{A}_j\Big) \propto N(\log(\sigma_{jl}^2)|m_{\sigma,l}, v_{\sigma,l}^2) \prod_{w_i = j} p(\mathbf{x}_i|\boldsymbol{\tau}_j).$$

- For those input vectors $\mathbf{x}_i$ with $w_i = j$, we have $x_{ik} \sim N(\cdot|\mathbf{a}_{jk}\boldsymbol{\eta}_i + \mu_{jk}, \sigma_{jk}^2)$, where $\mathbf{a}_{jk}$ is the kth row of $\mathbf{A}_j$. This indicates that the elements of $\mathbf{x}_i$ are independent. The posterior inference for $\mathbf{A}_j$ directly follows from the setting in the previous section.

- Assuming that $w_i = j$, we sample $\boldsymbol{\eta}_i$ from $N_r(\cdot|\mathbf{m}_{\eta_i}, \mathbf{V}_{\eta_i})$, where

$$\mathbf{V}_{\eta_i} = [\mathbf{B}_j' \boldsymbol{\Lambda}_j^{-1} \mathbf{B}_j + \mathbf{A}_j' \boldsymbol{\Sigma}_j^{-1} \mathbf{A}_j + \mathbf{I}_r]^{-1},$$
$$\mathbf{m}_{\eta_i} = \mathbf{V}_{\eta_i} \Big[\mathbf{B}_j' \boldsymbol{\Lambda}_j^{-1} (\mathbf{y}_i - \boldsymbol{\nu}_j) + \mathbf{A}_j' \boldsymbol{\Sigma}_j^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_j) \Big].$$

Acknowledgments

The authors would like to thank the editors and three anonymous referees for their constructive comments and suggestions on the original version of this paper. This work has been supported in part by the Natural Science Foundation of China (No. 61070239) and in part by the Office of Naval Research under contract number N00014-11-1-0688.