Flexible online multivariate regression with variational Bayes and the matrix-variate Dirichlet process

Victor Meng Hwee Ong and David J. Nott*

Department of Statistics and Applied Probability
National University of Singapore
6 Science Drive 2, Singapore, 117546

Taeryon Choi

Department of Statistics
Korea University
Republic of Korea

Ajay Jasra

Department of Statistics and Applied Probability
National University of Singapore
6 Science Drive 2, Singapore, 117546

Abstract. Flexible density regression methods, in which the whole distribution of a response vector changes with the covariates, are very useful in some applications. A recently developed technique of this kind uses the matrix-variate Dirichlet process as a prior for a mixing distribution on a coefficient in a multivariate linear regression model. The method is attractive for the convenient way that it allows borrowing strength across different component regressions and for its computational simplicity and tractability. The purpose of the present article is to develop fast online variational Bayes approaches to fitting this model, and to investigate how they perform compared to MCMC and batch variational methods in a number of scenarios.

1. Introduction. Flexible regression modeling of multivariate data is an active research topic in data sciences, with a wide range of applications. Recent technological developments allow us to collect data with increasing volume and speed, making real time or on-line analyses attractive. While a number of on-line regression procedures exist to identify nonlinear relationships between predictors and response variables for real time data analysis, little has been done for on-line multivariate regression modeling when the whole distribution of the response is modelled flexibly.

In this paper, we consider a fast and flexible method for on-line multivariate regression modeling using variational Bayesian sequential updating algorithms. The model we consider is based on the matrix-variate Dirichlet process of [37] (hereafter MatDP). A MatDP is a Dirichlet process having a matrix-variate normal base measure, and following [37] we use the MatDP as a prior on a mixing distribution for

2010 Mathematics Subject Classification. Primary: 62F15, 62G08; Secondary: 62J99.
Key words and phrases. Bayesian nonparametrics, matrix-variate Dirichlet process, mixture model, online regression, variational Bayes.

* Corresponding author: David J. Nott.
a coefficient matrix in multivariate linear regression with a radial basis. The model is attractive for the computational simplicity and convenience that comes from the use of the ordinary Dirichlet process, and for the ability to borrow strength among regressions for different response variables through the matrix-variate normal base prior. Further extensions to handle multivariate supervised learning problems were considered in [38]. There are many related approaches to the method of [37] for density regression with a univariate response using a Dirichlet process mixture of regression models. These include [27] for wavelet models in functional clustering, [29] for classification, [14] for generalized linear models and [23] for latent factor regression for functional and longitudinal data, among others.

In the works mentioned above for the MatDP and related univariate density regression methods, familiar Markov chain Monte Carlo (MCMC) methods are employed for computation. However, the computational burden of using MCMC methods is heavy for large data sets. The main contribution of the present work is to develop fast and real time implementations of flexible multivariate regression models with the MatDP prior using sequential variational approximations. Our method adapts the variational sequential updating and greedy search (VSUGS) algorithm of [36] for this task. The VSUGS algorithm was motivated as a variational version of the previous SUGS algorithm [34] for Dirichlet process mixture models.

The development of fast computational methods for complex Bayesian nonparametric models has been a very active area of recent research. An important early paper is [3] where a batch variational algorithm for fitting Dirichlet process mixture models was developed. In the on-line setting, some recent contributions include [33] and [8] who consider on-line algorithms for the hierarchical Dirichlet process, and various methods inspired by the stochastic variational inference framework of [15] (for example, [32]). [19] consider a fast on-line approach to fitting high-dimensional correlated data with a model incorporating some Bayesian nonparametric components; their method is a variational Bayes algorithm which is similar in approach to methods originally developed by [28]. [22] consider online approaches to fitting semiparametric regression models in the variational Bayes framework.

The next section describes the matrix-variate Dirichlet process mixture model that is considered throughout the rest of the article. In Section 3, a batch variational algorithm for the model is derived and then Section 4 discusses the VSUGS online algorithm which is able to work efficiently for very large datasets. Section 5 discusses predictive inference and a novel localized adjustment approach giving improved performance. Section 6 considers an application to weakly informative prior selection, Section 7 considers predictive performance of the methods in some benchmark data sets and Section 8 gives some concluding discussion.

2. Matrix-variate Dirichlet process mixture model. We consider the matrix-variate Dirichlet process mixture model of [37]. Specifically, let $y_i, i = 1, \ldots, n$ denote a collection of observed $m$-dimensional response vectors and $x_i, i = 1, \ldots, n$ denote corresponding $p$-dimensional vectors of covariates. A common flexible way to model the mean in a multivariate regression for the responses involves using some basis expansion where, denoting the $j$th element of $y_i$ by $y_{ij}$,

$$E(y_{ij}) = \beta_{0,j} + \sum_{r=1}^{N} \beta_{r,j} E_r(x_i)$$

(1)
where $E_r(x)$, $r = 1, \ldots, N$ are basis functions and $\beta_j = (\beta_{0j}, \ldots, \beta_{Nj})^T$ are coefficients, $j = 1, \ldots, m$. In motivating their approach, [37] discuss such a basis expansion, and consider setting $N = n$ and $E_r(x) = K(x, x_r)$ where $K(\cdot, \cdot)$ is a kernel function, so that the number of basis terms equals the number of observations. Here we will be concerned with an online implementation of their approach where $n$ is not known beforehand, so we will make a fixed choice of both $N$ and the basis functions $E_r(x)$, $r = 1, \ldots, N$. We give more details about this later.

Write $\beta = [\beta_1, \ldots, \beta_m]$ for the $(N + 1) \times m$ matrix of regression coefficients and $E_i = (1, E_1(x_1), \ldots, E_N(x_i))^T$. If we assume i.i.d errors in the regression (1) we can write

$$y_i = \beta^T E_i + \epsilon_i$$

where the $\epsilon_i$ are the errors having mean 0 and covariance matrix $\tau \Sigma$ say where $\tau > 0$ is a scale parameter. The reason for parametrizing the covariance matrix in this way will become clear later when conjugate prior specifications are considered. Flexible multivariate regression approaches using basis expansions of this type have been considered by many authors. The innovation of [37] is to consider a model in which the coefficient $\beta$ varies randomly between observations. The distribution of this coefficient is estimated from the data, and is given a Dirichlet process prior with a matrix-variate normal distribution as the base measure. That is, the Dirichlet process puts all its mass on a countable collection of points so that the number of basis terms equals the number of observations will become clear later when conjugate prior specifications are considered. Flexible multivariate regression approaches using basis expansions of this type have been considered by many authors. The innovation of [37] is to consider a model in which the coefficient $\beta$ varies randomly between observations. The distribution of this coefficient is estimated from the data, and is given a Dirichlet process prior with a matrix-variate normal distribution as the base measure. That is, the Dirichlet process puts all its mass on a countable collection of points so that the number of basis terms equals the number of observations.

Precisely, the model is

$$Y_i | E_i, \tilde{\beta}_i, \Sigma \sim N(\tilde{\beta}_i^T E_i, \tau \Sigma)$$

$$\tilde{\beta}_i | Q \sim Q$$

$$Q | \alpha, M \sim DP(\alpha, M)$$

where $DP(\alpha, M)$ denotes the Dirichlet process with precision parameter $\alpha$ and base measure $M$. The base measure $M$ in the model is chosen to be a matrix-variate normal distribution $N_{N+1,m}(0, \Omega \otimes \Sigma)$. An $s \times t$ random matrix $Z$ has a matrix-variate normal distribution $N_{s,t}(C, V \otimes W)$, where $C$ is an $s \times t$ matrix and $V$ and $W$ are $s \times s$ and $t \times t$ covariance matrices respectively, if its density takes the form

$$p(Z) = (2\pi)^{-st/2} |V|^{-t/2} |W|^{-s/2} \exp \left( \text{tr} \left( -\frac{1}{2} V^{-1} (Z - C) W^{-1} (Z - C)^T \right) \right).$$

In our model following [37], $\Omega$ is diagonal, $\Omega = \text{diag}(\omega_1, \ldots, \omega_{N+1})$, where $\omega_i \sim IG(a_i, b_i)$ with $a_i$ and $b_i$ known. Also, $\Sigma$ is inverse-Wishart with degrees of freedom $\nu$ and scale matrix $S$. $\tau$ is given an inverse gamma prior $IG(a_\tau, b_\tau)$ with $a_\tau$ and $b_\tau$ known.

The Dirichlet process puts all its mass on a countable collection of points so we can rewrite the model in the following way. Let $\{\tilde{\beta}_i\}_{i=1}^\infty$ be the distinct values appearing in the sequence $\{\tilde{\beta}_i\}_{i=1}^\infty$ with the $\tilde{\beta}_i$ indexed according to their order of occurrence in $\{\tilde{\beta}_i\}_{i=1}^\infty$. We let $\delta_i$ be an integer valued variable with $\delta_i = j$ if $\tilde{\beta}_i = \beta_j$. Write $\delta_{1:t} = (\delta_1, \ldots, \delta_t)^T$. Using the Pólya urn representation for the
Dirichlet process we can rewrite the model in the form

\[ Y_i | E_i, \beta, \Sigma, \delta_i \sim N(\beta^T_i E_i, \tau \Sigma) \]

\[ p(\delta, \beta) = p(\delta)p(\beta) \quad (3) \]

where \( p(\delta_{1:n}) = p(\delta_{1:n}|\alpha) = \prod_{i=1}^{n} p(\delta_i|\delta_{1:i-1}, \alpha) \), \( p(\beta) = \prod_{i=1}^{\infty} p(\beta_i) \) with \( p(\beta_i) \) the matrix-variate normal density \( N_{N+1,m}(0, \Omega \otimes \Sigma) \), the priors on \( \Omega \) and \( \Sigma \) are the same as before and the conditional densities \( p(\delta_i|\delta_{1:i-1}, \alpha) \) are defined by (using similar notation to [37])

\[ p(\delta_i = j|\delta_{1:i-1}, \alpha) = \left\{ \begin{array}{ll} \frac{n_j^{(i)}}{\alpha + j - 1} & j \in \{1, \ldots, n_i\} \\ \frac{\alpha}{\alpha + i - 1} & j = n_i + 1 \end{array} \right. \]

where \( n_j^{(i)} \) is the number of \( \delta_k, k < i \) equal to \( j \) and \( n_i \) is the number of distinct \( \beta_k \) appearing up to time \( i - 1 \). For the purpose of developing our fast online variational approximation algorithm we will use a truncated Dirichlet process mixture model based on a finite symmetric Dirichlet prior [20]. In this model the sequence \( \{\beta_i\}_{i=1}^{\infty} \) is truncated to \( \{\beta_i\}_{i=1}^{T} \) where \( T \) is the truncation point and

\[ p(\delta_i = j|\delta_{1:i-1}, \alpha) = \left\{ \begin{array}{ll} \frac{n_j^{(i)} + \alpha/T}{\alpha + i - 1} & j \in \{1, \ldots, n_i\} \\ \frac{\alpha}{\alpha + i - 1} & j = n_i + 1 \end{array} \right. \quad (4) \]

This is the model we discuss in what follows. Unlike the truncated stick-breaking representation, the finite symmetric Dirichlet prior is known to have the advantage that cluster labels are interchangeable, and its empirical performance does not degrade as \( T \) increases [20].

3. Variational inference. Consider a Bayesian model with parameter \( \xi \), prior \( p(\xi) \) and likelihood \( p(y|\xi) \). Variational Bayes computational methods [35, 18, 1, 25] attempt to approximate the posterior density \( p(\xi|y) \) by a more tractable and manageable variational density \( q(\xi) \), belonging to a convenient family. The choice of \( q(\xi) \) within the approximating family is usually made by minimizing the KL divergence between \( p(\xi|y) \) and \( q(\xi) \). It can be shown that

\[ \log p(y) = \int \log \left( \frac{p(\xi)p(y|\xi)}{q(\xi)} \right) q(\xi)d\xi + \int \log \left( \frac{q(\xi)}{p(\xi|y)} \right) q(\xi)d\xi \quad (5) \]

where \( p(y) = \int p(\xi)p(y|\xi) \ d\xi \). The first and second terms on the RHS of (5) are the variational lower bound \( \mathcal{L} \) (so-called because it forms a lower bound on \( \log p(y) \)) and the KL divergence between \( q(\xi) \) and \( q(\xi|y) \), respectively. From (5), it is clear that minimizing the KL divergence is equivalent to maximizing \( \mathcal{L} \). For further background see the references above.

Now, suppose that \( \xi \) can be partitioned into \( J \) subvectors, \( \xi_1, \ldots, \xi_J \). In variational Bayes, an approximating family for the posterior is considered where \( q(\xi) \) is assumed to factorize as \( \prod_{j=1}^{J} q(\xi_j) \). For each of the factors \( q(\xi_j) \), the lower bound is maximized with the other factors held fixed by choosing \( q(\xi_j) \) as

\[ \hat{q}(\xi_j) \propto \exp \left\{ E_{-\xi_j} \log p(y|\xi)p(\xi) \right\} \quad (6) \]

where \( E_{-\xi_j} \) denotes an expectation with respect to \( \prod_{i \neq j} q(\xi_i) \). Expression (6) is the basis of a blockwise gradient descent algorithm for maximizing \( \mathcal{L} \) where an initial choice is made for the factors and then each factor is updated in turn with the others fixed at current values until convergence.
One useful application of the variational approach is to approximate the posterior distribution of parameters in Bayesian nonparametric models. It is well known that there is usually no direct way to compute the posterior distribution in these models and that MCMC sampling methods for such models can be difficult and computationally expensive. These considerations motivated [3] to consider a mean-field variational inference algorithm for Dirichlet process mixture models. Their approach can be implemented for the model of Section 2, since the approach of [37] is based on an ordinary Dirichlet process mixture model, and we do implement such an approach later in our examples. Since this is a straightforward application of the algorithm of [3] we do not give further details of their method here. However, we develop an alternative batch variational Bayes algorithm which is also described in the next section. The algorithm of [3] is based on the stick breaking representation of the Dirichlet process; our alternative batch variational Bayes algorithm (like the later sequential algorithm of Section 4) is based on the Polya urn representation with the unknown mixing distribution integrated out. Although the alternative batch algorithm involves some further approximations, the purpose of developing this method is that it gives a batch algorithm similar to our later online approach, and provides another reference for comparison for the performance of the online algorithm where how much performance is lost through the sequential updating mechanism can be better understood. Also, many of the updating steps in the online algorithm are simple modifications of the corresponding steps for the batch algorithm.

3.1. Batch mean field updates for global parameters. We work with the model (3). For the MatDP mixture model, variational inferences for the parameters \( \beta_{1:T}, \Sigma, \tau, \omega_{1:N}, \delta_{1:n} \) are required. Define \( \theta = (\beta_{1:T}, \Sigma, \tau, \omega_{1:N})^T \) and \( \theta_j = (\beta_j, \Sigma, \tau, \omega_{1:N})^T \) for \( j = 1, \ldots, T \). In deriving approximate mean field updates we consider a slight expansion of the model (3). This will be helpful when discussing the online case later, since in our model expansion the variational posterior has the same form as the prior leading to a natural online implementation. In the expanded model the prior on \( \beta_i \) is changed from \( N_{N+1,m}(0, \Omega \otimes \Sigma) \) to \( N_{N+1,m}(M_i, \Omega_i \otimes \Sigma) \) with \( \Omega_i = (\Omega^{-1} + C_i)^{-1} \) where \( M_i \) and \( C_i \) are known matrices. Letting \( \beta = (\beta_1, \ldots, \beta_T) \), we consider the following factorization for the variational posterior distribution:

\[
q(\theta, \delta_{1:n}) = q(\beta, \Sigma)q(\tau)q(\omega_{1:N+1})q(\delta_{1:n}).
\]

\( q(\delta_i = j) \) will be denoted by \( q_{ij} \). In this subsection we give the mean field updates for all factors except for \( q(\delta_{1:n}) \), which is considered in the next subsection. Technical details of the derivations are found in Appendix A.

For \( \beta \), we recognize the form of \( q(\beta, \Sigma) \) as being \( q(\beta, \Sigma) = q(\Sigma)q(\beta|\Sigma) \) where \( q(\Sigma) \) is inverse Wishart, and \( q(\beta|\Sigma) = \prod_{j=1}^{T} q(\beta_j|\Sigma) \) with \( q(\beta_j|\Sigma) = N_{N+1,m}(\hat{\beta}_j, V_j^{-1} \otimes \Sigma) \),

\[
\hat{\beta}_j = V_j^{-1} \left( E_q(\Omega^{-1}) + C_j \right) M_j + E_q(\tau^{-1}) \sum_{i=1}^{n} q_{ij} E_i E_i^T
\]

and

\[
V_j = \left( E_q(\Omega^{-1}) + C_j + E_q(\tau^{-1}) \sum_{i=1}^{n} q_{ij} E_i E_i^T \right).
\]
For Σ, \( q(\Sigma) = IW(\tilde{\nu}, \tilde{S}) \) where \( \tilde{\nu} = \nu + n \) and
\[
\tilde{S} = S + \sum_{j=1}^{T} \left( E_q(y_i^{-1}) \left( \sum_{i=1}^{n} q_{ij} y_i y_i^T + M_j^T (E_q(\Omega^{-1}) + C_j) M_j - \beta_j^T V_j \beta_j \right) \right).
\]

For τ, \( q(\tau) = IG(\hat{a}_\tau, \hat{b}_\tau) \) where \( \hat{a}_\tau = a_\tau + \frac{mn}{T} \) and
\[
\hat{b}_\tau = b_\tau + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{T} \left( \frac{1}{2} \sum_{j=1}^{T} q_{ij} (y_i - \hat{\beta}_j^T E_i)^T E_q(\Sigma^{-1})(y_i - \hat{\beta}_j^T E_i) + m E_i^T V_j^{-1} E_i. \right)
\]

Lastly, for \( \omega = (\omega_1, \ldots, \omega_{N+1}) \), \( q(\omega) = \prod_{i=1}^{N+1} q(\omega_i) \). Each \( q(\omega_i) = IG(\hat{a}_i, \hat{b}_i) \) where \( \hat{a}_i = a_i + mT/2 \) and
\[
\hat{b}_i = b_i + \frac{1}{2} \sum_{j=1}^{T} \left( (\hat{\beta}_{j,i} - M_{j,i}) E_q(\Sigma^{-1})(\hat{\beta}_{j,i} - M_{j,i})^T + m \omega_{ij} \right),
\]
where \( \hat{\beta}_{j,i} \) is the \( i \)th row of \( \hat{\beta}_j \) and \( \omega_{ij} \) is the \( i \)th diagonal element of \( V_j^{-1} \).

### 3.2. Batch mean field update for local parameters.

We now factorize \( q(\delta_{i:n}) \) as \( \prod_{i=1}^{n} q(\delta_i) \) and consider approximate mean field updates for \( q(\delta_i), i = 1, \ldots, n \). Using (6), for each \( \delta_i \), we get
\[
q(\delta_i) \propto \exp \left( E_q \left\{ \log p(\delta_i | \delta_{\neq i}, \alpha) \right\} + E_q \left\{ \log(y_i | \theta, \delta_i) \right\} \right).
\]
where \( \delta_{\neq i} \) denotes \( \delta_{1:n} \) with \( \delta_i \) omitted. Making the approximation
\[
\exp \left( E_q \left\{ \log p(\delta_i | \delta_{\neq i}, \alpha) \right\} \right) \approx E_q \left\{ p(\delta_i | \delta_{\neq i}, \alpha) \right\}
\]
we have
\[
q(\delta_i) \propto E_q \left\{ p(\delta_i | \delta_{\neq i}, \alpha) \right\} \exp \left( E_q \left\{ \log(y_i | \theta, \delta_i) \right\} \right).
\]
If \( i \geq T \), we approximate further \( E_q \left\{ p(\delta_i | \delta_{\neq i}, \alpha) \right\} \) by
\[
\sum_{k \neq i} q_{ki} + \alpha/T, \quad \alpha + n - 1,
\]
with the case where \( i < T \) being handled by using the same expression but conditioning on \( \delta_i \leq i \). This approximation is obtained by reordering so that the \( i \)th observation is last, taking an expectation in (4) and then restoring the constraint associated with the original ordering by conditioning on \( \delta_i \leq i \) if \( i \leq T \). Note that because we order atoms according to their order of occurrence it must be the case that \( \delta_i \leq i \) for \( i \leq T \). To get an expression for our approximate mean field update it remains to evaluate \( E_q \left\{ \log(y_i | \theta_j) \right\} \) which is
\[
E_q \left\{ \log(y_i | \theta_j) \right\} = \frac{-m}{2} \log(2\pi) - \frac{m}{2} E_q \left\{ \log(2\pi) \right\} - \frac{1}{2} E_q \left\{ \log(S) \right\} \]
\[
= \frac{-1}{2} \left\{ E_q \left\{ \frac{1}{\tau} \right\} + E_q \left\{ y_i^T \Sigma^{-1} y_i - 2E_j^T \beta_j \Sigma^{-1} y_i + E_j^T \beta_j \Sigma^{-1} \beta_j^T E_j \right\} \right\}
\]
\[
= \frac{-m}{2} \log(2\pi) - \frac{m}{2} \left\{ \log(\hat{\beta}_\tau) - \psi(\hat{\beta}_\tau) \right\} \]
\[
= \frac{1}{2} \left\{ -\psi_m \left( \frac{\hat{\nu}}{2} \right) - m \log(2) + \log(\tilde{S}) \right\} - \frac{1}{2} \left( \frac{\hat{\nu} y_i^T (\tilde{S})^{-1} y_i}{b_\tau} \right)
\]
\[
- 2\hat{\nu} E_i^T \hat{\beta}_j (\tilde{S})^{-1} y_i + m E_i^T \hat{\beta}_j E_i + (E_i^T \hat{\beta}_j)^T (\nu (\tilde{S})^{-1} (E_i^T \hat{\beta}_j)).
\]
4. **VSUGS for MatDP mixture model.** The VSUGS algorithm, proposed by [36], is an online learning procedure for fast fitting of Dirichlet process mixture models. It uses the variational approximation framework to improve the SUGS algorithm [34]. The VSUGS algorithm is especially useful for large datasets as computing the full variational batch update or using MCMC might be computationally infeasible. The framework of the VSUGS procedure is as follows. Following [36], we consider an approximation to the posterior $p(\delta_{1:t-1}, \theta_{1:T}|y_{1:T})$ of the form

$$
\prod_{i=1}^{T} q_{i-1}(\delta_i) \prod_{j=1}^{T} q_{i-1}(\theta_j).
$$

The algorithm starts at $\hat{q}_1(\delta_1) = 1$, $\hat{q}(\theta_1) = p(\theta_1|y_1, \delta_1 = 1)$. Then, at time $i$, we use $\hat{q}_{i-1}(\theta)$ and $\hat{q}_{i-1}(\delta_{1:i-1})$ as a prior for processing the data point $y_i$. Then for a certain fixed choice of $\hat{q}_i(\delta_i)$ the mean field update for $\theta$ reduces to the following approximation of $p(\theta|y_i)$:

$$
\hat{q}_i(\theta) \propto \hat{q}_{i-1}(\theta) \prod_{j=1}^{T} \exp (\hat{q}_i(\delta_i = j) E_q \{ \log p(y_i|\theta_j) \}). \tag{8}
$$

For the assignment variables $\delta_i$, we follow [36] and choose

$$
\hat{q}_i(\delta_i = j) \propto r_{ij} \int \hat{q}_{i-1}(\theta_{\delta_i}) p(y_i|\theta_{\delta_i}) d\theta_{\delta_i}, \tag{9}
$$

for $j \in \{1, ..., \min(i, T)\}$ where $T$ is the pre-specified truncation point for the number of mixture components and

$$
r_{ij} = \left\{ \begin{array}{ll}
\sum_{k=1}^{i-1} \hat{q}_{i-1}(\delta_k = j) + \alpha/T & j \in \{1, \ldots, \min(i - 1, T)\}, \\
\alpha/(1 - \min(i - 1, T)/T) & j = \min(i - 1, T) + 1.
\end{array} \right. \tag{10}
$$

One property of the VSUGS procedure is that (9) splits the likelihood contribution from the $i$th observation among the mixture components. This deviates from the original SUGS algorithm [34] which uses a “hard” allocation to mixture components. In the case of conjugate priors, the VSUGS algorithm retains the computational advantages of the original SUGS algorithm. See [36] for further details.

4.1. **Sequential update of variational parameters for $\tau, \Sigma$ and $\Omega$.** In the batch update of the global parameters the expectations of $\tau^{-1}$, $\Sigma^{-1}$ and $\Omega^{-1}$ with respect to $q(\tau)$, $q(\Sigma)$ and $q(\Omega)$ respectively are required. For an online algorithm like VSUGS, these expectations change when a new data point enters. In order to use (8), it is required to replace the expectation of $\tau^{-1}, \Sigma^{-1}$ and $\Omega^{-1}$ with $E_{q}(\tau^{-1}), E_{q}(\Sigma^{-1})$ and $E_{q}(\Omega^{-1})$ respectively, where $E_{q}$ represents the variational expectation at time $i$. Following the derivation of the batch updates, our corresponding online learning update for the variational parameters is $q_i(\beta_j) \sim N_{q+1,m}(\hat{\beta}_j^{(i)}(i), (V_j^{(i)})^{-1} \otimes \Sigma)$, $q_i(\Sigma) \sim IW(\nu^{(i)}, S^{(i)})$, $q_i(\tau) \sim IG(\alpha^{(i)}, \beta^{(i)})$, where

$$
\hat{\beta}_j^{(i)} = \left( V_j^{(i-1)} \right)^{-1} \left( V_j^{(i-1)} \hat{\beta}_j^{(i-1)} + \frac{\alpha^{(i-1)}}{b_j^{(i-1)}} \hat{q}_i(\delta_i = j) E_i y_i T \right),
$$

$$
V_j^{(i)} = V_j^{(i-1)} + \frac{\alpha^{(i-1)}}{b_j^{(i-1)}} \hat{q}_i(\delta_i = j) E_i E_i^T,
$$

$$
\hat{\Sigma}_{ij}^{(i)} = \frac{\hat{q}_i(\delta_i = j) E_i}{\frac{\alpha^{(i-1)}}{b_j^{(i-1)}} + \frac{\alpha^{(i-1)}}{b_j^{(i-1)}} \hat{q}_i(\delta_i = j) E_i E_i^T},
$$

$$
\hat{\beta}_{ij}^{(i)} = \left( V_j^{(i-1)} \right)^{-1} \left( V_j^{(i-1)} \hat{\beta}_{ij}^{(i-1)} + \frac{\alpha^{(i-1)}}{b_j^{(i-1)}} \hat{q}_i(\delta_i = j) E_i y_i \right),
$$

$$
\hat{\Sigma}_{ij}^{(i)} = \frac{\hat{q}_i(\delta_i = j) E_i}{\frac{\alpha^{(i-1)}}{b_j^{(i-1)}} + \frac{\alpha^{(i-1)}}{b_j^{(i-1)}} \hat{q}_i(\delta_i = j) E_i E_i^T},
$$

$$
\hat{\Sigma}_{jj}^{(i)} = \frac{\hat{q}_i(\delta_i = j) E_j}{\frac{\alpha^{(i-1)}}{b_j^{(i-1)}} + \frac{\alpha^{(i-1)}}{b_j^{(i-1)}} \hat{q}_i(\delta_i = j) E_i E_i^T}.
$$


\[ \nu(i) = \nu + i. \]

\[ S(i) = S(i-1) + \sum_{j=1}^{T} \left\{ \frac{a_j^{(i-1)}}{b_j^{(i-1)}} \hat{q}_j(\delta_i = j)y_i y_i^T + \left( \hat{\beta}^{(i-1)}_j V_j^{(i-1)} S_j^{(i-1)} - \left( \hat{\beta}^{(i)}_j V_j^{(i)} S_j^{(i)} \right) \right) \right\}, \]

\[ a_j^{(i)} = a_r + \frac{im}{2}, \]

\[ b_j^{(i)} = b_r^{(i-1)} + \frac{1}{2} \sum_{j=1}^{T} \hat{q}_j(\delta_i = j) \{ (y_i - (\hat{\beta}^{(i)}_j V_j^{(i)} E_i)^T \nu^{(i)}(S^{(i)})^{-1}(y_i - (\hat{\beta}^{(i)}_j V_j^{(i)} E_i)) + mE_i^T(V_j^{(i)} - 1)) \}. \]

As the second term on the RHS of (8) does not include any terms for \( \omega_1, \ldots, \omega_{N+1} \), there is no online learning required for these parameters. At every step of the online VSUGS algorithm, we continue using the batch update for \( q_i(\omega_{1:N+1}) \). For the full algorithm, we refer to Algorithm 1.

### 4.2. Sequential VSUGS type update for the \( \delta_i \)

Suppose we assimilate observations sequentially and at step \( i - 1 \) we have a variational posterior distribution of the form

\[ q_{i-1}(\beta, \Sigma) q_{i-1}(\tau) q_{i-1}(\omega) q_{i-1}(\delta) \]

where \( q_{i-1}(\beta, \Sigma) = q_{i-1}(\Sigma) \prod_{j=1}^{T} q(\beta_j | \Sigma) \) with \( q_{i-1}(\Sigma) \) being \( IW(\nu^{(i-1)}_1, S^{(i-1)}_1) \), \( q_{i-1}(\beta_j | \Sigma) \) being \( N_{N+1,m}((\hat{\beta}^{(i-1)}_j, \Omega^{(i-1)}_j) \otimes \Sigma) \) and \( q_{i-1}(\tau) = IG(\alpha^{(i-1)}_r, b_r^{(i-1)}) \). Also \( q_{i-1}(\omega) = \prod_{j=1}^{N+1} q_{i-1}(\omega_j) \) with \( q_{i-1}(\omega_j) \) being \( IG(a^{(i-1)}_j, b^{(i-1)}_j) \) and \( q_{i-1}(\delta) = \prod_{j=1}^{T} q(\delta_j) \). Using the VSUGS approximation, we take

\[ \hat{q}_j(\delta_i = j) \propto r_{ij} \int p(y_i | \beta_j^T E_i, \tau \Sigma) q_{i-1}(\tau) q_{i-1}(\beta_j | \Sigma) q_{i-1}(\delta) d\beta_j d\Sigma d\tau \]

where

\[ r_{ij} = \begin{cases} \frac{\Sigma_{i-1}^{(i-1)} \hat{q}_{i-1}(\delta_i = j) + \alpha/T}{\alpha(1 - \min(i - 1, T)/T)} & j \in \{1, \ldots, \min(i - 1, T)\} \\ \frac{\alpha(1 - \min(i - 1, T)/T)}{\alpha + 1} & j = \min(i - 1, T) + 1 \end{cases} \]

The integral in (12) can be evaluated (see Appendix B)

\[ \int p(y_i | \beta_j^T E_i, \tau \Sigma) q_{i-1}(\tau) q_{i-1}(\beta_j | \Sigma) q_{i-1}(\delta) d\beta_j d\Sigma d\tau \]

\[ = \int (2\pi)^{-\frac{m}{2}} q_{i-1}(\tau) |\Omega_j^{(i-1)}|^{-\frac{m}{2}} \frac{1}{\tau} E_i E_i^T + (\Omega_j^{(i-1)})^{-1} \frac{1}{\tau} \frac{|\sigma|^{(i-1)} + 1/2}{2^{(\nu^{(i-1)}+1)/2}} \left| \frac{\sigma^{(i-1)}}{2^{(\nu^{(i-1)}+1)/2}} \right| \left| \frac{\nu^{(i-1)} + 1}{\tau} y_i y_i^T + (\hat{\beta}^{(i-1)}_j V_j^{(i-1)} \Omega_j^{(i-1)})^{-1} \hat{\beta}^{(i-1)}_j \right| d\tau \]

\[ = \frac{1}{\tau} E_i E_i^T + (\Omega_j^{(i-1)})^{-1} \frac{\nu^{(i-1)} + 1}{2} \int d\tau \]

\[ = \frac{1}{\tau} E_i E_i^T + (\Omega_j^{(i-1)})^{-1} \frac{\nu^{(i-1)} + 1}{2} \]
where $\hat{\beta}_j = \left(\frac{1}{2} E_i E_i^T + \Omega_j^{(i-1)}\right)^{-1} \left(\frac{1}{2} E_i y_i^T + \Omega_j^{(i-1)} - \hat{\beta}_j^{(i-1)}\right)$. This last integral does not seem to be easily computable analytically. It is an expectation with respect to $q_{i-1}(\tau)$, and if this distribution is concentrated around the mean it is reasonable to make the approximation $\int f(\tau) q_{i-1}(\tau) d\tau = f(\bar{E}_{i,i-1}(\tau))$ for functions $f(\tau)$ and where we have written $E_{i,i-1}(\tau) = \int \tau q_{i-1}(\tau) d\tau$. Using this approximation here we get that the integral is approximately

$$
\left(\frac{2\pi}{\mu_{i-1}}\right)^{-m/2} \left|\Omega_j^{(i-1)}\right|^{-m/2} |\mu_{i-1}^{(i-1)} E_i E_i^T + (\Omega_j^{(i-1)})^{-1}|^{-m/2} \text{ (15)}
$$

$$
\frac{|S^{(i-1)}|^{\nu(i-1)/2} \Gamma_m \left(\frac{\nu(i-1)+1}{2}\right)}{2^{\nu(i-1)m/2} \Gamma_m (\mu(i-1)/2)} 2^{(\nu(i-1)+1)m} \left|S^{(i-1)}\right|^{\nu(i-1)+1} \left|\mu_{i-1}^{(i-1)} E_i E_i^T + (\Omega_j^{(i-1)})^{-1}\right|^{-\nu(i-1)+1}
$$

where $\hat{\beta}_j = \left(\mu_{i-1}^{(i-1)} E_i E_i^T + (\Omega_j^{(i-1)})^{-1}\right)^{-1} \left(\mu_{i-1}^{(i-1)} E_i y_i^T + (\Omega_j^{(i-1)})^{-1} \beta_j^{(i-1)}\right)$ and the expectation of $\tau^{-1}$ with respect to $q_{i-1}(\tau)$ is denoted $\mu_{i-1}^{(i-1)}$.

5. **Posterior predictive inference.** Suppose we are given a new input vector $x_0$ and wish to predict the response vector $y_0$. Write $E_0 = (1, E_1(x_0), \ldots, E_N(x_0))^T$. The posterior predictive distribution of $y_0$ can be evaluated as

$$
p(y_0|x_0, y_{1:n}) = \sum_{j=1}^{T} \int p(y_0|y_{1:n}, \tau, \Sigma, \beta_{1:T}, \delta_0 = j)p(\tau, \Sigma, \beta_{1:T}|y_{1:n}) d\beta_1 d\tau d\Sigma.
$$

Assuming $n > T$, we replace $p(\delta_0 = j|y_{1:n})$ with $r_{n+1,j}$ as defined in (12). Also, replacing $p(\tau, \beta_j, \Sigma|y_{1:n})$ with the corresponding variational posterior $q_n(\tau, \beta_j, \Sigma)$, the predictive density becomes

$$
p(y_0|E_0, y_{1:n}) = \sum_{j=1}^{T} r_{n+1,j} \int \int p(y_0|y_{1:n}, \tau, \Sigma, \beta_j, \delta_0 = j)q_n(\tau)q_n(\beta_j)q_n(\Sigma) d\beta_j d\tau d\Sigma. \text{ (16)}
$$

The integral in (16) evaluates to a multivariate t-distribution. So an approximate posterior predictive density is obtained as a mixture of multivariate t-densities. For more details, including the parameters of the multivariate-t mixture components, see Appendix C.

5.1. **Regression-type adjustment for improving predictive inference.** One advantage of using the matrix-variate Dirichlet process approach to flexible regression is that avoiding covariate dependence in the mixing weights greatly simplifies computation, something that we have exploited here for implementing an online algorithm. However, this does place a greater burden on the mean functions in the regression mixture components to model the response distribution in a flexible way. Here we consider a method for improving predictive performance of the fitted model, borrowing an idea from the literature on regression adjustment methods for approximate Bayesian computation [2, 4, 5, 6]. The idea below is given in equation (4.1) of [6].
Suppose we wish to consider prediction of a new response $y^* = (y_1^*, \ldots, y_m^*)^T$ to be observed with corresponding covariate $x^* = (x_1^*, \ldots, x_m^*)^T$. Write $N_k(x^*)$ for the $k$ nearest neighbours of $x^*$ among the observed covariates $\{x_1, \ldots, x_n\}$. We write the corresponding values of $(x, y)$ as $(x_{i1}, y_{i1}), \ldots, (x_{ik}, y_{ik})$ so that $i_1, \ldots, i_k$ denote the indices of the covariates in $N_k(x^*)$. In our fitted regression model, we
write \( \hat{F}_j(.|x) \) for the marginal distribution function of the \( j \)th component of the response in the fitted model at \( x \), and \( \hat{F}_j^{-1}(.|x) \) for its inverse where it is assumed this exists.

If the fitted model is correct, \( \hat{F}_j(y_{ir}|x_ir) \) is uniform on \([0, 1]\), and \( \hat{F}_j^{-1}(\hat{F}_j(y_{ir}|x_ir)|x^*) \) has the distribution \( \hat{F}_j(|x^*) \). So if we set

\[
y^a_r = (y^a_{r1}, \ldots, y^a_{rm})^T = (\hat{F}^{-1}_1(\hat{F}(y_{i1}|x_{ir})), \ldots, \hat{F}^{-1}_m(\hat{F}(y_{im}|x_{ir})|x^*))^T
\]

\( r = 1, \ldots, k \), then marginally \( y^a_{r1}, \ldots, y^a_{rm} \) is a sample from \( \hat{F}_j(\cdot|x^*) \) (if the regression model is correct).

The sample \( y^a_r, r = 1, \ldots, k \) can be used to do approximate predictive inference. The advantage of this method compared to using \( \hat{F}_j(\cdot|x^*) \) directly is that by using, in effect, quantile residuals locally around \( x^* \) to define the particles \( y^a_r \) we are able to adjust for any local misfit of the regression model. This can result in improved predictive inference. Note that by transforming the particles \( y_{ir} \) component-wise we are not guaranteed to preserve the correct multivariate dependence structure in the fitted model at \( x^* \), but if the copula of the fitted distribution changes only slowly with \( x \) over the neighbourhood used the effects of this approximation are minor.

6. Application to weakly informative prior selection. As an application of our proposed methodology, we consider flexible approximation of prior predictive densities as a function of a prior hyperparameter value based on data simulated under a model, when these prior predictive densities are not analytically tractable. Approximating such predictive densities is useful for prior choice. In the application considered here we make use of the way that the MDP mixture model is able to approximate the whole response distribution flexibly. In the next section we will look more closely at the quality of point predictions of the online algorithm compared to those obtained by batch VB and MCMC approaches.

Consider a statistical model \( p(y|\xi) \) for data \( y \) with parameter \( \xi \). Suppose we have a class of priors \( p(\xi|\lambda) \) where \( \lambda \) is a hyperparameter value to be chosen. We also suppose that there is a value \( \lambda_0 \) for \( \lambda \) that has already been chosen tentatively as representing our best current prior knowledge of \( \xi \). For the purpose of sensitivity analysis, we may wish to define a prior that is less informative than \( p(\xi|\lambda_0) \), and this might be particularly useful in the case where the information brought by the prior and likelihood seem to be contradictory. [9] considered defining the amount of information in a prior \( p(\xi|\lambda) \) relative to \( p(\xi|\lambda_0) \) through the idea of prior-data conflict. The notion of weakly informative priors formalized in [9] was inspired by previous work of [11].

Since the idea of [9] is built on the idea of checking for prior-data conflict, this needs to be understood first. Prior-data conflict occurs where the prior puts all its mass out in the tails of the likelihood. A way of testing for prior-data conflict which modifies a suggestion of [7] will be considered here, following [10]. Their idea is that a minimal sufficient statistic value \( S \) determines the likelihood, so we can check if the observed likelihood is in conflict with the prior by seeing whether the observed value of the sufficient statistic \( S_{obs} \) say lies out in the tails of its prior predictive distribution. A \( p \)-value for checking for conflict with the prior \( p(\xi|\lambda) \) can be computed as

\[
p(S_{obs}, \lambda) = P(p(S|\lambda) \leq p(S_{obs}|\lambda)),
\]

\( r = 1, \ldots, k \).
where $S \sim p(S|\lambda)$ and $p(S|\lambda) = \int p(S|\xi)p(\xi|\lambda)d\xi$ is the prior predictive distribution of $S$. If a non-trivial sufficient statistic does not exist it may be reasonable to choose an asymptotically sufficient statistic such as the maximum likelihood estimator or some approximation to it. Note that $p(S_{\text{obs}}, \lambda)$ is calculating the probability that a random draw from $p(S|\lambda)$ has lower density than the value of $S_{\text{obs}}$ and it is small if $S_{\text{obs}}$ lies out in the tails of $p(S|\lambda)$. The above prior-data conflict check can be modified in various ways – for more details see [10].

To use this notion of prior-data conflict checking to define how informative the prior $p(\xi|\lambda)$ is relative to $p(\xi|\lambda_0)$ [9] consider $S$ generated randomly under $p(S|\lambda_0)$ and ask whether for data generated in such a way does doing the analysis under $p(\xi|\lambda_0)$ rather than $p(\xi|\lambda_0)$ result in a reduction of the frequency of prior-data conflicts. The occurrence of a conflict is defined by choice of a certain cutoff for a $p$-value such as (18). It is possible to consider various modifications of the basic idea considering uniformity of reduction of levels of conflict over different $p$-value cutoffs, see [9] for more details.

Following the ideas of [10] and [9], [24] propose modifying a regression adjustment approach used in the approximate Bayesian computation (ABC) literature to approximate prior predictive distributions $p(S|\lambda)$ for many different $\lambda$ in a computationally thrifty way when $S$ may be expensive to compute. In particular, they consider the method of [5], which modifies a suggestion of [2], to generate approximate samples from prior predictive densities $p(S|\lambda)$ and then use these samples for the required computations. This approach is much more computationally efficient than generating a large number of replications of $S$ at each value of $\lambda$ independently for every value $\lambda$ of interest on a grid, say. The method starts by generating values $\lambda_i$, $i = 1, \ldots, n$ from a pseudo-prior $p(\lambda)$. Then values $(\xi_i, S_i)$, $i = 1, \ldots, n$ are generated for $(\xi, S)$ from $p(\xi|\lambda)p(S|\xi)$ where $S$ is a minimal sufficient statistic or some asymptotically sufficient statistic. [24] modify the ABC with regression adjustment method in [5] by reversing the usual role of the parameters and the summary statistics where these methods are used in the ABC context. They fit a regression model with

$$S_i = \mu(\lambda_i) + \sigma(\lambda_i)\epsilon_i$$

(19)

where the $\epsilon_i$ are i.i.d errors with zero mean and variance one and $\mu(\lambda)$ and $\sigma(\lambda)$ are flexible mean and standard deviation functions. [5] parametrize $\mu(\cdot)$ and $\sigma(\cdot)$ using neural networks. After fitting the model to the data to obtain estimates $\hat{\mu}(\lambda)$ and $\hat{\sigma}(\lambda)$, a sample of $p(S|\lambda)$ can be obtained approximately by considering the fitted mean for the regression model plus the empirical residuals. The empirical residual for the $i$th point is $\hat{\epsilon}_i = \hat{\sigma}(\lambda_i)^{-1}(S_i - \hat{\mu}(\lambda_i))$, and using such empirical residuals together with the fitted model at $\lambda$ gives

$$S^*_i(\lambda) = \hat{\mu}(\lambda) + \hat{\sigma}(\lambda)\{\hat{\sigma}(\lambda_i)^{-1}(S_i - \hat{\mu}(\lambda_i))\}$$

(20)

as an approximate sample from $p(S|\lambda)$ if the regression model is correct. Based on the approximate sample $S^*_i(\lambda)$, they use a kernel estimate to approximate $p(S|\lambda)$. Let this kernel estimate be $\hat{p}(S|\lambda)$. Next, suppose that $S^*_j$, $j = 1, \ldots, n$ are draws from $p(S|\lambda_0)$. Then a particle approximation to the distribution of $p(S, \lambda)$ for $S \sim p(S|\lambda_0)$ is given by the values $\hat{P}(S^*_1, \lambda), \ldots, \hat{P}(S^*_n, \lambda)$ where

$$\hat{P}(S^*_j, \lambda) = \frac{1}{n} \sum_{i=1}^{n} I(\hat{p}(S^*_i(\lambda)|\lambda) \leq \hat{p}(S^*_j(\lambda)|\lambda))$$
The distribution of the $p$-value can be used to determine whether $p(\xi|\lambda)$ is weakly informative relative to $p(\xi|\lambda_0)$ or not.

We propose using our approach to assess weak informativity of alternative priors compared to a base prior, similar to the above. However, instead of using the ABC with regression adjustment (19), we propose fitting a matrix-variate Dirichlet process mixture model with $S$ as response and $\lambda$ as predictors. In applying the MDP prior approach we also employ the regression adjustment method of Section 5.1 to obtain approximate samples from $p(S|\lambda)$ at any desired value of $\lambda$. Kernel estimates of $p(S|\lambda)$ are then constructed as for the approach of [24] and the procedure above followed for approximating the distribution of conflict $p$-values for $S$ generated from $p(S|\lambda_0)$. As observed in [24], high accuracy is not needed in the regression calculations; the regression calculations are simply a screening computation, and once a candidate value of $\lambda$ is chosen for a weakly informative prior then for the single finaly chosen value we can generate a large sample from the prior predictive distribution and see whether our approximate calculations were good enough.

6.1. Analysis of a logistic regression example. We consider a bioassay example from [26] which is also analysed in [12], [9] and [24]. In this dataset, four groups of five animals were exposed to different level of doses ($x_i$) and the number of death ($y_i$) were recorded. Following [24] and [9], we consider a logistic regression setup. It is assumed that the covariate has been transformed to log scale and centred and scaled as in [12]. The model is $y_i \sim \text{Bin}(5,p_i)$ where $\text{logit}(p_i) = c_0 + c_1 x_i$. We assume that the priors for $c_0$ and $c_1$ are independent and follow Gaussian distributions with zero mean and variances $\sigma_0^2$ and $\sigma_1^2$ respectively. For our base prior, we consider $\sigma_0 = 10$ and $\sigma_1 = 2.5$.

[9] consider the exact sufficient statistics $(y_1, y_2, y_3, y_4)$ for analysis. [24] consider using the posterior mode $(\hat{c}_0, \hat{c}_1)$ for a prior with $\sigma_0 = \sigma_1 = 10$ as an approximation.
to the MLE but which unlike the MLE will exist even in degenerate cases. They consider the MLE for the dimension reduction that it brings and as a generic choice applicable in situations where a non-trivial minimal sufficient statistic doesn’t exist. For the statistic $S$ used to define the conflict check in the definition of weak informativity, they use a transformation of $(c_0, c_1)$ to the fitted probabilities $\hat{p}_2$ and $\hat{p}_3$ at $x_2$ and $x_3$ respectively. The reasons for this are discussed further in [24]. That is, our approximate sufficient statistic is $(\hat{p}_2, \hat{p}_3)$ where $\hat{p}_i = 1/(1 + \exp(-c_0 + c_1 x_i))$ for $i = 2, 3$. Note that, because of the discreteness of the data, strictly the distribution of this statistic is also discrete but continuity may be used as a reasonable approximation when the number of different possible values is large and we do this here. Note also that in the kernel density estimation we ignore any boundary effects due to the bounded support of the statistics.

To use our methodology to investigate weak informativity with respect to the base prior in this example we proceed as follows. First, we generate 400,000 values of $(\sigma_0, \sigma_1)$ from a pseudo prior which is uniform distribution on $[0.1, 10] \times [0.1, 20]$. We label these values as $\sigma^{(i)} = (\sigma_0^{(i)}, \sigma_1^{(i)})$ for $i = 1, \ldots, 400,000$. For each $\sigma^{(i)}$, we generate $c_0^{(i)}$ and $c_1^{(i)}$ from their respectively prior distribution and then compute the probability $(\hat{p}_1^{(i)}, \hat{p}_2^{(i)}, \hat{p}_3^{(i)}, \hat{p}_4^{(i)})$. We use these probabilities to generate $(\hat{y}_1^{(i)}, \hat{y}_2^{(i)}, \hat{y}_3^{(i)}, \hat{y}_4^{(i)})$ from their respective binomial distribution.

Let $\hat{p}^{(i)} = (\hat{p}_2^{(i)}, \hat{p}_3^{(i)})^T$ and $K(\sigma^{(i)}, \sigma^{(j)}) = \exp\{-||\sigma^{(i)} - \sigma^{(j)}||^2/2\kappa^2\}$, where $||.||$ is the Euclidean norm and $\kappa^2$ is the mean of the euclidean distance among 5000 random samples drawn from $\sigma^{(1)}, \ldots, \sigma^{(400,000)}$. [37] propose a similar choice of the kernel hyperparameter $\kappa^2$ and it is verified to be effective in their experimental analysis. We fit the matrix-variate Dirichlet process mixture model with $\hat{p}^{(i)}$ as our response vector and the basis functions $(1, K(\sigma^{(i)}, \sigma^{(1)}), \ldots, K(\sigma^{(i)}, \sigma^{(N)}))$ as our covariates. We set $a = 100$ and $T = 4$. For our prior, we set $a_3 = 5$, $b_3 = 0.5$, $a_4 = 5$, $b_4 = 0.5$, $S = I_2 + \frac{1}{2}I_2I_2^T$, $\nu = 3$ and $M_1, \ldots, M_T, C_1, \ldots, C_T$ as zero matrices.

We first run Algorithm 1 to fit the model. For each $\lambda_\sigma$ on a 100x100 regular grid on $[0.1, 10] \times [0.1, 20]$, we take 1000 nearest neighbours from the set $\{\sigma^{(i)}\}_{i=1,\ldots,400,000}$ using the kmeans function in MATLAB. Then, we use these 1000 nearest neighbour to estimate the corresponding $(\hat{p}_2, \hat{p}_3)^T$ using the regression adjusted approach proposed in Section 5.1. Note that if we were to generate 1000 samples directly for the prior predictive for each of our 10,000 grid points directly this would increase the number of prior predictive simulations and the computational effort by an order of magnitude. The collection of $\lambda_\sigma$ covers the support of the hyperprior and each $\lambda_\sigma$ is considered as an alternative prior for comparison with the base prior. We follow the suggestion from [9] to measure the degree of weak informativity of an alternative prior. That is, we let $p_\gamma$ be the $\gamma\%$ quantile of the conflict $p$-value distribution for the base prior. Under the alternative prior, let $q_\gamma$ be the probability of a conflict $p$-value which is less than or equal to $p_\gamma$. The degree of weak informativity $\zeta_\gamma$ is defined as

$$\zeta_\gamma = \begin{cases} 0 & q_\gamma > p_\gamma \\ 1 - q_\gamma/p_\gamma & q_\gamma \leq p_\gamma \end{cases}$$

Figure 1(a) plots the degree of weak informativity $\zeta_\gamma$ for all $\lambda_\sigma$ when $\gamma = 0.05$. We observe that the plot is very similar to Figure 2 in [24]. From the plot, it seems that $\sigma_0 = 2$ and $\sigma_1 = 7.5$ is a suitable choice for a weakly informative prior. This conclusion agrees with the parameter choice of [24]. As mentioned earlier, the approximate regression calculations are simply screening calculations where
high accuracy is not needed since the quality of the final answer can be checked. Figure 1(b) shows a comparison of the estimated distribution of the conflict \( p \)-values based on regression (grey) compared to one based on direct simulation from the prior predictive at the finally chosen \( \lambda \) (black). In the lower tail, which is what matters for declaring the existence of any conflict and defining weak informativity, the two distributions agree very well, and the approximate regression calculations have successfully allowed us to identify a suitable weakly informative prior. Note that standard procedures such as Gibbs sampler proposed in [37] are not suitable for use in this application with the matrix-variate DP prior model as we need to generate a large number of data points from the hyperprior to obtain good estimates of the prior predictive distributions \( p(S|\lambda) \) and so a method is needed that is able to handle large datasets.

7. Empirical comparisons of predictive performance. In this section, we focus on the predictive performance of the various computational approaches to fitting the model. We consider the Gibbs sampler [37] and three different versions of the variational procedure. The first variational approach, which we call VB (Stick breaking), is the method of [3]. The second variational method, which we call VB (Pólya urn) is the batch variational method discussed in Section 3. In the third variational approach, we consider the VSUGS approach discussed in Section 4.

In our experiment, the predictors are standardized to have zero mean and unit variance with respect to the training set. We fixed the number of iterations for the Gibbs sampler to be 25,000, of which the first 15,000 will be discarded as burn in. For the remaining 10,000 iterations of the Gibbs sampler, we retain every 10th realization of the parameters. We also fixed the number of iterations for the variational approach using batch updates to 100. For the online variational approach, we first initialize the variational parameters of the assignment variables on a relatively small number of data points using the batch update and then run Algorithm 1.

We measure the performance of the various approaches by considering their root mean square error (RMSE) and mean absolute percentage error (MAPE). Let \( \hat{y}_1, \ldots, \hat{y}_m \) be our target response values and \( \tilde{y}_1, \ldots, \tilde{y}_m \) be their respective fitted value. The error indicators are defined as

\[
RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (\tilde{y}_i - \hat{y}_i)^2}
\]

and

\[
MAPE = \frac{1}{m} \sum_{i=1}^{m} \left| \frac{\tilde{y}_i - \hat{y}_i}{\tilde{y}_i} \right|.
\]

Similar performance measures are used in [37].

In our analysis, we compute the RMSE and MAPE both in-sample (for the training set) and out-of-sample (for the test set). Although we are mostly interested in out-of-sample predictive performance, looking at in-sample measures of fit can also be useful here where we are comparing several computational approximations for the same posterior; measures of in-sample fit can be revealing about differences in the quality of posterior approximation even if out-of-sample predictive performance is similar for the different methods. For the RMSE and MAPE of in-sample predictions, they are constructed as follows. For the Gibbs sampler, at each retained
MCMC realization $r$, $r = 1, \ldots, 1000$, we have a cluster allocation for each $y_1, \ldots, y_n$. Suppose let’s say that for a particular data point $y_i$, the allocation at the $r$ realization is the $j$th component. Then our corresponding regression coefficient estimate would then be $\beta_{i,r}^* = \beta_{j,r}^T$, where $\beta_{1,r}, \ldots, \beta_{T,r}$ are the $r$th MCMC coefficient matrix realizations. Then, our in-sample fitted value for $y_i$ is estimated as

$$\left\{ \sum_{r=1}^{R} \beta_{i,r}^* \right\}^T E_i.$$ 

For all variational procedures, for each $y_i$, we have the corresponding variational posterior probability of the assignments as well as the posterior mean. We use the weighted sum of these coefficient matrices according to the posterior probability to compute an in-sample fitted value.

Out of sample fitted values are obtained from the posterior predictive distributions of the respective procedures. For the Gibbs sampler, we will use the predictive distribution from equation (7) in [37]. For the VSUGS procedures, we use the posterior predictive mean to fit each $y_i$. Details of the posterior predictive mean can be found in Section 5. For prediction accuracy of the test set, we also consider the adjusted VSUGS approach, which is to use the VSUGS with the regression-type adjustment in Section 5.1.

7.1. Energy Data. In this example, we consider the energy efficiency data created by [31]. This dataset is available at http://archive.ics.uci.edu/ml/datasets. [31] studied the effect of eight input variables (relative compactness, surface area, wall area, roof area, overall height, orientation, glazing area, glazing area distribution) on two output variables, namely heating load (HL) and cooling load (CL), of residential buildings. The dataset contains 768 instances. We randomly select 100 data points as the test set and use the remaining 668 data points as the training set.

The settings for fitting our model are as follows. We set the number of basis function, $N$, at 200 and use the same kernel discussed in Section 6.1. Our choice of $\alpha$ is 3, which is set by rounding off the average of the 10,000 iterations of $\alpha$ retained from the Gibbs sampler. For comparison of performance, we run both the Gibbs sampler and the variational procedures with a fixed $\alpha = 3$. For the hyperparameters of the priors, we set $a_{\tau} = 5$, $b_{\tau} = 0.5$, $a_i = 20$, $b_i = 0.5$, $S = I_2 + \frac{1}{2}I_21_2^T$, $\nu = 3$ and $M_1, \ldots, M_T, C_1, \ldots, C_T$ as zero matrices. We also set the maximum number of possible components as $T = 10$. For the variational approach with batch update, we initialize each assignment variables randomly to one of the $T$ components and set its variational probability to one. For the matrix VSUGS approach, we initialize the variational parameters for the first 200 assignment variables using the VB(Pólya urn).

| Method                  | RMSE  | MAPE  | $y_1$ | $y_2$ | $y_3$ | Mean | $y_1$ | $y_2$ | $y_3$ | Mean | Time (mins) |
|-------------------------|-------|-------|-------|-------|-------|------|-------|-------|-------|------|-------------|
| Gibbs Sampler           | 0.0474| 0.0450| 0.0462| 0.0685| 0.0667| 0.0636| 141   |
| VB (Stick Breaking)     | 0.1853| 0.2150| 0.2091| 0.2163| 0.2469| 0.2163| 18    |
| VB (Pólya urn)          | 0.1579| 0.1562| 0.1570| 0.1954| 0.2806| 0.2380| 18    |
| Matrix VSUGS           | 0.3387| 0.3160| 0.3273| 0.4527| 0.5176| 0.4851| 3     |

Table 1. In-sample accuracy of various approaches for the energy efficiency data.
Table 2. Prediction accuracy of various approaches for test set for the energy efficiency data

| Method                  | $y_1$   | $y_2$   | Mean   | $y_1$   | $y_2$   | Mean   |
|-------------------------|---------|---------|--------|---------|---------|--------|
| Gibbs Sampler           | 0.5736  | 0.5881  | 0.5809 | 0.7427  | 0.8055  | 0.7741 |
| VB (Stick Breaking)     | 0.3930  | 0.4911  | 0.4421 | 0.5731  | 0.8347  | 0.7039 |
| VB (Polya urn)          | 0.4005  | 0.5000  | 0.4503 | 0.4784  | 0.5038  | 0.4911 |
| Matrix VSUGS            | 0.4038  | 0.4882  | 0.4460 | 0.5364  | 0.6140  | 0.5752 |
| Adjusted VSUGS          | 0.2558  | 0.3329  | 0.2943 | 0.3399  | 0.4687  | 0.4043 |

Measures of in-sample fit and computation times are presented in Table 1. The two batch VB methods have similar in-sample fits, but the Gibbs sampling and VSUGS approaches produce quite different results. Table 3 considers out-of-sample predictive accuracy. Table 2 shows that all three variational approaches without the regression-type adjustment perform similarly. The RMSE and MAPE of the Gibbs sampler is higher than the rest. Table 2 also shows that using regression-type adjustment significantly improves prediction accuracy with respect to both RMSE and MAPE.

7.2. Robot Arm Data. In this subsection, we analyse the performance of our proposed algorithm on the benchmark robot arm data. The dataset relates to an inverse dynamics problem for a seven degrees-of-freedom SARCOS anthropomorphic robot arm, and is available from [www.gaussianprocess.org/gpml/data](http://www.gaussianprocess.org/gpml/data). The dataset has 21 covariates and 7 responses and has training and test sets of sizes 44,448 and 4,449 respectively. The 21 covariates consist of 7 joint positions, 7 joint velocities and 7 joint accelerations and the 7 responses consist of 7 joint torques.

We follow the same procedure used in the energy dataset for our settings. The value of $\alpha$ is set to 12. For the hyperparameters of the prior, we set $a = 5$, $b = 0.5$, $a_1 = 20$, $b_1 = 0.5$, $S = I_7 + \frac{1}{5}1_71_7^T$, $\nu = 8$ and $M_1, \ldots, M_T, C_1, \ldots, C_T$ as zero matrices. We also set the maximum number of possible components as $T = 10$. For the matrix VSUGS approach, we initialize the variational parameters of the first 500 data points using VB (Polya urn).

Table 3. In-sample accuracy of various approaches for the robot arm data.

| Method                  | $y_1$   | $y_2$   | $y_3$   | $y_4$   | $y_5$   | $y_6$   | $y_7$   | Mean   |
|-------------------------|---------|---------|---------|---------|---------|---------|---------|--------|
| Gibbs Sampler           | 0.1439  | 0.1338  | 0.1154  | 0.0884  | 0.1615  | 0.1615  | 0.1698  | 0.1304 |
| VB (Stick Breaking)     | 0.2058  | 0.1827  | 0.1492  | 0.1512  | 0.1917  | 0.2086  | 0.1521  | 0.1773 |
| VB (Polya urn)          | 0.1887  | 0.1976  | 0.1704  | 0.1297  | 0.2042  | 0.2121  | 0.1803  | 0.1774 |
| Matrix VSUGS            | 0.5297  | 0.4665  | 0.3994  | 0.4125  | 0.4531  | 0.4630  | 0.4097  | 0.4463 |
| Adjusted VSUGS          | 0.6158  | 0.5692  | 0.5054  | 0.5601  | 0.5902  | 0.8916  | 0.4842  | 0.5975 |

The performance of the in-sample fits is presented in Table 3, for fitting to a subsample of size 2,000. Corresponding out-of-sample fits for test set subsample of size 500 are shown in Table 4. We also initialize the variational parameters of the first 2,000 data points using VB (Polya urn) and then run Algorithm 1 on the full training dataset of size 44,448. As the dataset is large, we do not use the Gibbs sampler or the variational procedures using batch updates for the full dataset. The results of out-of-sample predictive performance are presented in Table 5. It is clear that there is a significant prediction accuracy improvement in terms of RMSE and MAPE when using the adjusted VSUGS.
Table 4. Prediction accuracy of various approaches for the robot arm data.

| Method                | y1  | y2  | y3  | y4  | y6  | y6  | y7  | Mean |
|-----------------------|-----|-----|-----|-----|-----|-----|-----|------|
| **RMSE**              |     |     |     |     |     |     |     |      |
| Gibbs Sampler         | 0.4099 | 0.3636 | 0.3404 | 0.3638 | 0.3598 | 0.3867 | 0.3395 | 0.3662 |
| VB (Stick Breaking)   | 0.5524 | 0.4909 | 0.4547 | 0.4225 | 0.5227 | 0.4653 | 0.4037 | 0.4732 |
| VB (Pólya urn)        | 0.4323 | 0.4195 | 0.3983 | 0.3851 | 0.4410 | 0.3916 | 0.3762 | 0.4063 |
| Matrix VSUGS          | 0.4198 | 0.3650 | 0.3375 | 0.3684 | 0.3737 | 0.3867 | 0.3490 | 0.3714 |
| Adjusted VSUGS        | 0.3513 | 0.3105 | 0.2869 | 0.2910 | 0.3249 | 0.3479 | 0.2741 | 0.3124 |
| **MAPE**              |     |     |     |     |     |     |     |      |
| Gibbs Sampler         | 1.0327 | 0.5987 | 1.5229 | 1.4137 | 1.0385 | 0.7330 | 1.2885 | 1.0897 |
| VB (Stick Breaking)   | 1.3711 | 0.7539 | 1.9556 | 1.0920 | 1.2136 | 1.1278 | 1.3825 | 1.2709 |
| VB (Pólya urn)        | 0.7587 | 0.6397 | 0.9774 | 0.9263 | 1.4217 | 1.1021 | 1.1947 | 1.0029 |
| Matrix VSUGS          | 0.9679 | 0.6600 | 1.8472 | 1.4600 | 1.2228 | 0.8428 | 1.4971 | 1.2140 |
| Adjusted VSUGS        | 0.7813 | 0.6097 | 1.7723 | 0.6627 | 1.1116 | 0.8206 | 1.0809 | 0.9770 |

Table 5. Prediction accuracy of various approaches using full training set for robot arm data.

| Method                | y1  | y2  | y3  | y4  | y6  | y6  | y7  | Mean |
|-----------------------|-----|-----|-----|-----|-----|-----|-----|------|
| RMSE                  |     |     |     |     |     |     |     |      |
| Matrix VSUGS          | 0.3930 | 0.3453 | 0.3665 | 0.3671 | 0.3402 | 0.3380 | 0.2923 | 0.3311 |
| Adjusted VSUGS        | 0.2505 | 0.2500 | 0.2315 | 0.1626 | 0.2541 | 0.2515 | 0.1785 | 0.2255 |
| MAPE                  |     |     |     |     |     |     |     |      |
| Matrix VSUGS          | 0.7204 | 0.4575 | 1.8183 | 0.7178 | 0.8687 | 0.7910 | 0.7220 | 0.8708 |
| Adjusted VSUGS        | 0.7204 | 0.4575 | 1.8183 | 0.7178 | 0.8687 | 0.7910 | 0.7220 | 0.8708 |

8. Discussion. In this article, we study variational computational methods for fitting the matrix-variate Dirichlet process mixture model of [37], extending the VSUGS approach by [36] for Dirichlet process mixtures of normal densities. The method we develop is computationally efficient, and especially useful as an alternative to MCMC for analysis of medium to large datasets. In order to increase prediction accuracy, we also propose a regression-type adjustment for improving predictive inference. The adjustment approach is shown to be useful in several real applications.

There are numerous ways that the accuracy of our method might be improved. Various strategies (e.g. [17], [30], [16], [15], [21], [20] and references therein) can be employed for the initialization of the variational approach using the batch update, in addition to different approximation schemes, and different settings for the maximum number of distinct components. For the fast on-line VB algorithm, we could consider the number of clusters in the MatDP mixture as a random variable with a variational distribution [17] as well as a hard truncation using the finite symmetric Dirichlet prior [20]. Alternatively, an additional prune and merge step proposed in [21] can be exploited. For functional data analysis, functional clustering is a natural extension of the MatDP mixture model to obtain smooth function representations in clusters for functional data. Motivated by the clustering property of Dirichlet process, this variant of the MatDP regression can be considered in future work.

A further limitation of the methodology lies in the use of conjugate priors, essential though they are for the computational development. In particular, the appearance of the covariance matrix $\Sigma$ in both the likelihood and the prior’s base measure $M$ may lead to undesirable statistical consequences. Fast methods which relax the requirement of conjugacy are a further topic for future research.

Appendix A - Variational Batch update. In the derivation of the update for each block $\gamma$ say we will write simply $E_q(\cdot)$ for the expectation with respect to the current variational posterior distribution $q$ with $\gamma$ integrated out and will not denote the dependence on the block explicitly in the notation. The meaning will be
clear from the context. We consider the mean field update for \((\beta, \Sigma)\) first. We have
\[
q(\beta, \Sigma) \propto \exp \left( E_q \left( \sum_{i=1}^{n} \sum_{j=1}^{T} I(\delta_i = j) \log p(y_i|\beta_j^T E_i, \tau \Sigma) + \sum_{j=1}^{T} \log p(\beta_j|\Omega, \Sigma) + \log p(\Sigma) \right) \right).
\]

Apart from constant terms not depending on \(\beta, \Sigma\) we have
\[
E_q(\log p(\Sigma)) = -\frac{\nu + m + 1}{2} \log |\Sigma| - \frac{1}{2} \text{tr}(S\Sigma^{-1})
\]
\[
E_q(\log p(\beta_j|\Omega, \Sigma)) = -\frac{N + 1}{2} \log |\Sigma|
\]
\[
- \frac{1}{2} \text{tr} \left( (E_q(\Omega^{-1}) + C_j)(\beta_j - M_j)\Sigma^{-1}(\beta_j - M_j)^T \right)
\]
\[
E_q \left( \sum_{i=1}^{n} \sum_{j=1}^{T} I(\delta_i = j) \log p(y_i|\beta_j^T E_i, \tau \Sigma) \right) = \sum_{i=1}^{n} \sum_{j=1}^{T} q_{ij} E_q(\log p(y_i|\beta_j^T E_i, \tau \Sigma))
\]

where (again apart from terms not depending on \(\beta, \Sigma\))
\[
E_q(\log p(y_i|\beta_j^T E_i, \tau \Sigma)) = -\frac{1}{2} \log |\Sigma| - \frac{1}{2} E_q(\tau^{-1})(y_i - \beta_j^T E_i)^T \Sigma^{-1}(y_i - \beta_j^T E_i).
\]

This gives (again up to an additive constant)
\[
\log q(\beta, \Sigma) = -\frac{(n + N + 2)m + \nu + 1}{2} \log |\Sigma| - \frac{1}{2} \text{tr}(S\Sigma^{-1}) - \frac{1}{2} \sum_{j=1}^{T} \text{tr} \left( (E_q(\Omega^{-1}) + C_j)(\beta_j - M_j)\Sigma^{-1}(\beta_j - M_j)^T \right)
\]
\[
- \frac{1}{2} E_q(\tau^{-1}) \sum_{i=1}^{n} \sum_{j=1}^{T} q_{ij} (y_i - \beta_j^T E_i)^T \Sigma^{-1}(y_i - \beta_j^T E_i).
\]

To simplify this, write
\[
\hat{\beta}_j = V_j^{-1} \left( E_q(\Omega^{-1}) + C_j + E_q(\tau^{-1}) \sum_{i} q_{ij} E_i y_i^T \right)
\]
where
\[
V_j = \left( E_q(\Omega^{-1}) + C_j + E_q(\tau^{-1}) \sum_{i} q_{ij} E_i E_i^T \right)
\]
and observe that
\[
\text{tr} \left( (E_q(\Omega^{-1}) + C_j)(\beta_j - M_j)^{-1}(\beta_j - M_j)^T \right) \\
+ E_q(\tau^{-1}) \sum_{i=1}^{n} q_{ij}(y_i - \beta_j^T E_i)^T \Sigma^{-1}(y_i - \beta_j^T E_i) \\
= \text{tr} \left( (\beta_j - M_j)^T (E_q(\Omega^{-1}) + C_j)(\beta_j - M_j) \Sigma^{-1} \right) \\
+ \text{tr} \left( E_q(\tau^{-1}) \sum_{i=1}^{n} q_{ij}(y_i - \beta_j^T E_i)(y_i - \beta_j^T E_i)^T \Sigma^{-1} \right) \\
= \text{tr}(\beta_j - \hat{\beta}_j)^T V_j (\beta_j - \hat{\beta}_j) \Sigma^{-1} + \text{tr}(M_j^T(E_q(\Omega^{-1}) + C_j)M_j \Sigma^{-1}) \\
+ \text{tr} \left( E_q(\tau^{-1}) \sum_{i=1}^{n} q_{ij} y_i \Sigma^{-1} \right) - \text{tr} \left( \hat{\beta}_j^T V_j \hat{\beta}_j \Sigma^{-1} \right).
\]

This means that up to an additive constant
\[
\log q(\beta, \Sigma) = \\
- \frac{(n + N + 2)m + \nu + 1}{2} \log |\Sigma| - \frac{1}{2} \sum_{j=1}^{T} \text{tr}( (\beta_j - \hat{\beta}_j)^T V_j (\beta_j - \hat{\beta}_j) \Sigma^{-1} ) \\
- \frac{1}{2} \text{tr} \left( \left( S + \sum_{j=1}^{T} \left( E_q(\tau^{-1}) \sum_{i=1}^{n} q_{ij} y_i y_i^T + M_j^T (E_q(\Omega^{-1}) + C_j) M_j - \hat{\beta}_j^T V_j \hat{\beta}_j \right) \right) \right)^{-1}.
\]

Then we recognize the form of \( q(\beta, \Sigma) \) as being \( q(\beta, \Sigma) = q(\Sigma)q(\beta|\Sigma) \) where \( q(\Sigma) \) is inverse Wishart,
\[
\text{IW} \left( \nu + n, S + \sum_{j=1}^{T} \left( E_q(\tau^{-1}) \sum_{i=1}^{n} q_{ij} y_i y_i^T + M_j^T (E_q(\Omega^{-1}) + C_j) M_j - \hat{\beta}_j^T V_j \hat{\beta}_j \right) \right)
\]
and \( q(\beta|\Sigma) = \prod_{j=1}^{T} q(\beta_j|\Sigma) \) with \( q(\beta_j|\Sigma) = N_{N+1,m}(\hat{\beta}_j, V_j^{-1} \otimes \Sigma) \).

Next, let’s consider the mean field update for \( q(\tau) \). We have
\[
q(\tau) \propto \exp(E_q \sum_{i=1}^{n} \sum_{j=1}^{T} I(\delta_i = j) \log p(y_i | \beta_j^T E_i, \tau, \Sigma) + \log p(\tau)).
\]

Apart from additive constants
\[
E_q(\log p(\tau)) = \log p(\tau) = -(a_\tau + 1) \log \tau - b_\tau / \tau
\]
and
\[
E_q \left( \sum_{i=1}^{n} \sum_{j=1}^{T} I(\delta_i = j) \log p(y_i | \beta_j^T E_i, \tau, \Sigma) \right) = \sum_{i=1}^{n} \sum_{j=1}^{T} q_{ij} E_q(\log p(y_i | \beta_j^T E_i, \tau, \Sigma))
\]
where
\[
E_q(\log p(y_i | \beta_j^T E_i, \tau, \Sigma)) = \frac{-nm}{2} \log \tau - \frac{1}{2\tau} E_q((y_i - \beta_j^T E_i) \Sigma^{-1}(y_i - \beta_j^T E_i)).
\]

Next,
\[
E_q((y_i - \beta_j^T E_i)^T \Sigma^{-1}(y_i - \beta_j^T E_i)) = E_q(E_q((y_i - \beta_j^T E_i)^T \Sigma^{-1}(y_i - \beta_j^T E_i))) \Sigma).
\]

To evaluate the inner conditional expectation, we use the following Lemma (Guptar and Nagar, p. 60).
Lemma 8.1. Suppose that $X \sim N_{p,q}(M, \Delta \otimes \psi)$. Let $A$ be a $p \times p$ matrix. Then
\[ E(X^TAX) = tr(\Delta A^T)\psi + M^TAM \]

Using Lemma 8.1 and some simple algebra we obtain
\[ E_q((y_i - \beta_j^TE_i)\Sigma^{-1}(y_i - \beta_j^TE_i)|\Sigma) = (y_i - \tilde{\beta}_j^TE_i)\Sigma^{-1}(y_i - \beta_j^TE_i) + mE_i^TV_j^{-1}E_i. \]
Hence
\[ E_q((y_i - \beta_j^TE_i)^T\Sigma^{-1}(y_i - \beta_j^TE_i)) = (y_i - \tilde{\beta}_j^TE_i)^TE_q(\Sigma^{-1})(y_i - \beta_j^TE_i) + mE_i^TV_j^{-1}E_i. \]
So apart from additive constants
\[ \log q(\tau) = -(a_\tau + \frac{nm}{2} + 1) \log \tau - \frac{1}{\tau} \left\{ b_\tau + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{T} q_{ij}(y_i - \tilde{\beta}_j^TE_i)^T E_q(\Sigma^{-1})(y_i - \beta_j^TE_i) \\
+ mE_i^TV_j^{-1}E_i) \right\}. \]
Hence we recognize that $q(\tau)$ is inverse gamma,
\[ IG(a_\tau + \frac{nm}{2}, b_\tau + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{T} q_{ij}(y_i - \tilde{\beta}_j^TE_i)^T E_q(\Sigma^{-1})(y_i - \beta_j^TE_i) + mE_i^TV_j^{-1}E_i). \]

Next we consider the variational update for $\omega$. We have that apart from additive constants
\[ \log q(\omega) = -\sum_{i=1}^{N+1} (a_i + 1) \log \omega_i - \sum_{i=1}^{N+1} \frac{b_i}{\omega_i} - \frac{mT}{2} \sum_{i=1}^{N+1} \log \omega_i \\
- \frac{1}{2} \sum_{i=1}^{N+1} \frac{1}{\omega_i} \sum_{j=1}^{T} E_q((\beta_{j,i} - M_{j,i})^T \Sigma^{-1}(\beta_{j,i} - M_{j,i})) \]
where $\beta_{j,i}$ and $M_{j,i}$ denote the $i$th rows of $\beta_j$ and $M_j$ respectively. Writing
\[ E_q((\beta_{j,i} - M_{j,i})^T \Sigma^{-1}(\beta_{j,i})) = E_q((\beta_{j,i} - M_{j,i})^T \Sigma^{-1}(\beta_{j,i})), \]
and noting that $\beta_{j,i}^T \Sigma \sim N(\hat{\beta}_{j,i}, \omega_{ij}'\Sigma)$ where $\hat{\beta}_{j,i}$ is the $i$th row of $\hat{\beta}_j$ and $\omega_{ij}'$ is the $i$th diagonal element of $V_j^{-1}$ we have
\[ E_q((\beta_{j,i} - M_{j,i})^T \Sigma^{-1}(\beta_{j,i})) = \hat{\beta}_{j,i} - M_{j,i})^T \Sigma^{-1}(\hat{\beta}_{j,i} - M_{j,i}) + \text{tr}(\Sigma^{-1}\omega_{ij}'\Sigma) \\
= (\hat{\beta}_{j,i} - M_{j,i})^T E_q(\Sigma^{-1})(\hat{\beta}_{j,i} - M_{j,i}) + \omega_{ij}'m. \]
So apart from additive constants
\[ \log q(\omega) = -\sum_{i=1}^{N+1} (a_i + mt/2 + 1) \log \omega_i \\
- \frac{1}{\omega_i}(b_i + \frac{1}{2} \sum_{j=1}^{T} ((\hat{\beta}_{j,i} - M_{j,i})^T E_q(\Sigma^{-1})(\hat{\beta}_{j,i} - M_{j,i}) + \omega_{ij}'m) \]
and so we recognize that $q(\omega) = \prod_{i=1}^{N+1} q(\omega_i)$ where $q(\omega_i)$ is inverse gamma,
\[ IG(a + mT/2, b + \frac{1}{2} \sum_{j=1}^{T} ((\hat{\beta}_{j,i} - M_{j,i})E_q(\Sigma^{-1})(\hat{\beta}_{j,i} - M_{j,i}) + m\omega_{ij}'), \]
(22)
Appendix B - Sequential update for $\delta_i$. To help evaluate the integral we use the following lemma.

**Lemma 8.2.** (a) Let $Z$, $V$, $W$ and $C$ be matrices with $Z \in \mathbb{R}^{s \times t}$, $V > 0 \in \mathbb{R}^{s \times s}$, $W > 0 \in \mathbb{R}^{t \times t}$ and $C \in \mathbb{R}^{s \times s}$. Then

$$\int \exp\left(-\frac{1}{2}(V^{-1}ZW^{-1}Z^T - 2V^{-1}CW^{-1}Z^T)\right) dZ = (2\pi)^{s/2}|V|^{s/2}|W|^{t/2}\exp\left\{\frac{1}{2}V^{-1}CW^{-1}C^T\right\}.$$  

(b) Suppose $\psi$, $A$ are matrices with $\psi > 0 \in \mathbb{R}^{m \times m}$ and $A > 0 \in \mathbb{R}^{m \times m}$. Let $a > 0$ be a constant. Then

$$\int |\psi|^{-(a+m+1)/2} \exp\left(-\frac{1}{2}A\psi^{-1}\right) d\psi = |A|^{-a/2}2^{a m/2}\Gamma_m(a/2)$$

where $\Gamma_m(x) = \pi^{m(m-1)/4} \prod_{i=1}^{m} \Gamma(x + (1 - i)/2)$ is the multivariate gamma function.

**Proof.** Part a) of the lemma follows easily from the fact that the integrand is an unnormalized matrix normal distribution. Using the fact that the integral of the corresponding normalized density is one and rearranging gives the result. Part b) follows in a similar way by noting that the integrand is an unnormalized inverse Wishart density.

Using Lemma 8.2 to help evaluate the integral we get

$$\int \rho(q_i|\beta_j E_i, \tau \Sigma) q_i^{-1}(\tau) q_i^{-1}(\Sigma) d\beta_j d\Sigma d\tau$$

$$= \int \left(2\pi\right)^{-m/2}|\Sigma|^{-1/2} q_i^{-1}(\tau) \exp\left\{-\frac{1}{2}y_i^T \Sigma^{-1} y_i\right\} \left(2\pi\right)^{-(N+1)m/2} \Omega_j^{(i-1)} \left|\Omega_j^{(i-1)}\right|^{-m/2} \exp\left(-\frac{1}{2}\Omega_j^{(i-1)} \beta_j^{(i-1)} \Sigma^{-1} \beta_j^{(i-1)}\right)$$

$$\int \exp\left\{\frac{1}{2} \left(\frac{1}{\tau} E_i^T \beta_j + (\Omega_j^{(i-1)})^{-1} \beta_j \Sigma^{-1} \beta_j^T\right)\right\} d\beta_j d\Sigma d\tau$$

where $\hat{\beta}_j = \left(\frac{1}{\tau} E_i^T + (\Omega_j^{(i-1)})^{-1}\right)^{-1} \left(\frac{1}{\tau} E_i^T y_i + (\Omega_j^{(i-1)})^{-1} \beta_j^{(i-1)}\right)$. Next, use part (a) of Lemma 8.2 and the explicit form for $q_i^{-1}(\Sigma)$ to get

$$\int \left(2\pi\right)^{-m/2} q_i^{-1}(\tau) \left|\Omega_j^{(i-1)}\right|^{-m/2} \frac{1}{\tau} E_i^T + (\Omega_j^{(i-1)})^{-1} \frac{1}{m/2} \frac{S_j^{(i-1)} \rho^{(i-1)} / 2}{2 \nu^{(i-1)} m/2 \Gamma_m(\nu^{(i-1)}/2)}$$

$$\int \left|\Sigma\right|^{-(\nu^{(i-1)+m+2)/2}} \exp\left(-\frac{1}{2} \left(S_j^{(i-1)} + \frac{1}{\tau} y_i y_i^T + (\beta_j^{(i-1)})^T \Omega_j^{(i-1)} (\beta_j^{(i-1)})^{-1}\right) d\Sigma d\tau.$$

$$- \beta_j^T \left(\frac{1}{\tau} E_i^T + (\Omega_j^{(i-1)})^{-1} \beta_j \Sigma^{-1}\right) d\Sigma d\tau.$$
Using part (b) of Lemma 8.2, this is equal to

\[
\int (2\pi)^{-\nu/2} |\tau|^{\nu/2} \Gamma_m((\nu+1)/2) 2^{(\nu+1)/2} \prod_j \left| S(j) + \frac{1}{\tau} y_j y_j^T (\Omega_j^{-1}) - \frac{1}{\tau} \right|^{\nu/2} d\tau.
\]

\[
= \frac{1}{\tau} E_i E_i^T (\Omega_j^{-1}) - \frac{1}{\tau} |\Omega_j^{-1}|^{-1/2} \frac{1}{\tau} E_i E_i^T + (\Omega_j^{-1})^{-1} \frac{1}{\tau}.
\]

Appendix C - Posterior predictive distribution. We now evaluate the integral in (16). Following the same steps to get (15), we have

\[
\int p(y_0 | y_{1:n}, \tau, \Sigma, \beta, \delta_0 = j) q_n(\tau) q_n(\beta_j) q_n(\Sigma) d\beta_j d\tau d\Sigma
\]

\[
= \left( \frac{2\pi}{\mu_{\tau^{-1}}^{(n)}} \right)^{-m/2} \left| \Omega_j^{(n)} \right|^{-m/2} \left| \mu_{\tau^{-1}}^{(n)} E_0 E_0^T + (\Omega_j^{(n)})^{-1} \right|^{-m/2}
\]

\[
\times \frac{2^{(\nu(n)+1)/m}}{2^{(\nu(n)/m/2)} \Gamma_m((\nu(n)/2))} \left| S^{(n)} \right|^{\nu(n)/2}
\]

\[
\times \left| S^{(n)} + \mu_{\tau^{-1}}^{(n)} y_0 y_0^T + \left( \beta_j^{(n)} \right)^T (\Omega_j^{(n)})^{-1} \beta_j^{(n)} - \beta_j^{(n)} (\mu_{\tau^{-1}}^{(n)} E_0 E_0^T + (\Omega_j^{(n)})^{-1} \beta_j^{(n)}) \right|^{-\nu(n)/2}.
\]

where \( \beta_j = \left( \mu_{\tau^{-1}}^{(n)} E_0 E_0^T + (\Omega_j^{(n)})^{-1} \right)^{-1} (\mu_{\tau^{-1}}^{(n)} E_0 y_0^T + (\Omega_j^{(n)})^{-1} \beta_j^{(n)}) \). We now show that after simplification of the term inside the determinant we obtain the multivariate t-distribution. We require here the matrix determinant lemma, which states the following.

Lemma 8.3. For any invertible matrix \( A \) and vector \( u \) and \( v \) we have

\[
|A + uv^T| = (1 + v^T A^{-1} u) |A|.
\]

Letting \( A = (\mu_{\tau^{-1}}^{(n)} E_0 E_0^T + (\Omega_j^{(n)})^{-1}) \), it is clear that

\[
\beta_j^{(n)} (\mu_{\tau^{-1}}^{(n)} E_0 E_0^T + (\Omega_j^{(n)})^{-1}) \beta_j
\]

\[
= (\mu_{\tau^{-1}}^{(n)} E_0 y_0^T + (\Omega_j^{(n)})^{-1} \beta_j^{(n)})^T A^{-1} A \beta_j
\]

\[
= (\mu_{\tau^{-1}}^{(n)} E_0 y_0^T + (\Omega_j^{(n)})^{-1} \beta_j^{(n)})^T A^{-1} E_0 y_0^T + \mu_{\tau^{-1}}^{(n)} y_0 E_0^T A^{-1} (\Omega_j^{(n)})^{-1} \beta_j^{(n)}
\]

\[
+ \mu_{\tau^{-1}}^{(n)} (\beta_j^{(n)})^T (\Omega_j^{(n)})^{-1} E_0 y_0^T + (\beta_j^{(n)})^T (\Omega_j^{(n)})^{-1} A^{-1} \beta_j^{(n)}.
\]
This gives us

\[
\begin{aligned}
S^{(n)} + \mu_{\tau-1}^{(n)} y_0 T + (\tilde{\beta}_j^{(n)})^T (\Omega_j^{(n)})^{-1} \tilde{\beta}_j^{(n)} - \tilde{\beta}_j^{(n)} (\mu_{\tau-1}^{(n)} E_0 E_0^T + (\Omega_j^{(n)})^{-1}) \tilde{\beta}_j^{(n)}
\end{aligned}
\]

\[
= S^{(n)} + \mu_{\tau-1}^{(n)} y_0 T + (\tilde{\beta}_j^{(n)})^T (\Omega_j^{(n)})^{-1} \tilde{\beta}_j^{(n)} - (\mu_{\tau-1}^{(n)})^2 y_0 E_0^T \Lambda^{-1} E_0 y_0 T
\]

\[
- \mu_{\tau-1}^{(n)} y_0 E_0^T \Lambda^{-1} \Omega_j^{(n)} \tilde{\beta}_j^{(n)} - \mu_{\tau-1}^{(n)} (\tilde{\beta}_j^{(n)})^T (\Omega_j^{(n)})^T \Lambda^{-1} E_0 y_0 T
\]

\[
- (\tilde{\beta}_j^{(n)})^T (\Omega_j^{(n)})^T \Lambda^{-1} \Omega_j^{(n)} \tilde{\beta}_j^{(n)} \frac{\nu(n)+1}{2}
\]

Let \( S_* = S^{(n)} - (\tilde{\beta}_j^{(n)})^T (\Omega_j^{(n)})^T \Lambda^{-1} \Omega_j^{(n)} \tilde{\beta}_j^{(n)} + (\tilde{\beta}_j^{(n)})^T (\Omega_j^{(n)})^T \Lambda^{-1} \tilde{\beta}_j^{(n)} \) (where we suppress dependence on \( j \) in the notation). Using Lemma 8.3, we have

\[
1 + \mu_{\tau-1}^{(n)} y_0 T S_*^{-1} y_0 (1 - \mu_{\tau-1}^{(n)} E_0^T \Lambda^{-1} E_0) - 2 \mu_{\tau-1}^{(n)} E_0^T \Lambda^{-1} \Omega_j^{(n)} \tilde{\beta}_j^{(n)} S_*^{-1} y_0 \frac{\nu(n)+1}{2} |S_*| \frac{\nu(n)+1}{2}
\]

As the term inside the absolute value is a quadratic function in \( y_0 \), the predictive distribution is a multivariate t-distribution. Letting \( A = \mu_{\tau-1}^{(n)} (1 - \mu_{\tau-1}^{(n)} E_0^T \Lambda^{-1} E_0) S_*^{-1} \), \( B = -2 \mu_{\tau-1}^{(n)} S_*^{-1} (\tilde{\beta}_j^{(n)})^T \Omega_j^{(n)} \Lambda^{-1} E_0 \) (again suppressing dependence on \( j \) in the notation), we have

\[
1 + \mu_{\tau-1}^{(n)} y_0 T S_*^{-1} y_0 (1 - \mu_{\tau-1}^{(n)} E_0^T \Lambda^{-1} E_0) - 2 \mu_{\tau-1}^{(n)} y_0 T S_*^{-1} (\tilde{\beta}_j^{(n)})^T \Omega_j^{(n)} \Lambda^{-1} E_0
\]

\[
= 1 + (y_0 + \frac{1}{2} A^{-1} B)^T A (y_0 + \frac{1}{2} A^{-1} B) - \frac{1}{4} B^T A A B
\]

\[
= 1 + (y_0 + \frac{1}{2} A^{-1} B)^T A (1 - \frac{1}{2} B^T A^{-1} B)^{-1} (y_0 + \frac{1}{2} A^{-1} B) \frac{1}{4} B^T A A B.
\]

Thus, the \( j \)th component in the expression for the posterior predictive density (16) is multivariate \( t \) with location \(-\frac{1}{2} A^{-1} B\) and scale \( \frac{1}{\nu(n)-m-1} \{ A (1 - \frac{1}{2} B^T A^{-1} B)^{-1} \}^{-1} \). Hence (16) is approximately a mixture of multivariate \( t \)-distributions.

**Appendix D - Variational Lower bound.** Next we compute the variational lower bound on \( \log p(y_{1:n}) \), which is defined as

\[
\mathcal{L}(q) = E_q \left\{ \log \left[ p\left( y_{1:n}, \beta_{1:T}, \delta_{1:n}, \tau, \omega_{1:(N+1)}, \Sigma \right) \right] \right\}
\]

\[
- E_q \left\{ \log \left[ q\left( \beta_{1:T}, \delta_{1:n}, \tau, \omega_{1:(N+1)}, \Sigma \right) \right] \right\}.
\]

Similar to the approach used to calculate the lower bound for the normal mixture model in [36], we approximate \( \mathcal{L}(q) \) recursively. Let \( \theta_i = (\hat{\beta}_i, \tau, \omega_{1:N+1}, \Sigma) \) and using
r_{ij} as an approximation to p(\delta_i|y_{i:t-1}), we have

\[ L(q) = E_{q_i} \{ \log(q_{i-1}(p_i, \tau, \omega, \nu; \Sigma))p(\delta_i|y_{i:t-1})p(y_i|\delta_i, \theta) \} - E_{q_i} \{ \log(q(\theta)q(\delta_i)) \]

\[ = E_{q_i} \left\{ \log(q_{i-1}(\tau)) + \log(q_{i-1}(\Sigma)) + \sum_{j=1}^{N+1} \log(q_{i-1}(\omega_j)) + \sum_{t=1}^{T} \log(q_{i-1}(\beta_t)) + \log(r_{ij}) + \log(y_i|\theta, \delta_i) \right\} - E_{q_i} \left\{ q_i(\tau) + \log(q_i(\Sigma)) + \sum_{j=1}^{N+1} \log(q_i(\omega_j)) + \sum_{t=1}^{T} \log(q_i(\beta_t)) + \sum_{t=1}^{T} \log(q_i(\beta_t)) + \log(q_i(\delta_i)) \right\}. \]

Suppressing the expectations with respect to \( q_i \), evaluating the terms involving \( \tau \) and \( \omega_j \) gives us

\[ E_{q_i} \{ \log(q_{i-1}(\tau)) \} - E_{q_i} \{ \log(q_i(\tau)) \} = (a_r^{(i)} - a_r^{(i-1)})\psi(a_r^{(i)}) - \log(\Gamma(a_r^{(i)})) + \log(\Gamma(a_r^{(i-1)})) + a_r^{(i-1)}(\log(b_r^{(i-1)}) - \log(b_r^{(i)})) + a_r^{(i)} \frac{b_r^{(i)} - b_r^{(i-1)}}{b_r^{(i)}}, \]

\[ E_{q_i} \{ \log(q_{i-1}(\omega_j)) \} - E_{q_i} \{ \log(q_i(\omega_j)) \} = (a_j^{(i)} - a_j^{(i-1)})\psi(a_j^{(i)}) - \log(\Gamma(a_j^{(i)})) + \log(\Gamma(a_j^{(i-1)})) + a_j^{(i-1)}(\log(b_j^{(i-1)}) - \log(b_j^{(i)})) + a_j^{(i)} \frac{b_j^{(i)} - b_j^{(i-1)}}{b_j^{(i)}}. \]

Since \( q_i(\Sigma) \) follows an inverse Wishart distribution with degrees of freedom \( \nu^{(i)} \) and scale matrix \( S^{(i)} \), we obtain

\[
E_{q_i} \{ \log(q_{i-1}(\Sigma)) \} = \frac{\nu^{(i-1)}}{2} \log |S^{(i-1)}| - \frac{\nu^{(i-1)}m}{2} \log(2) - \log \Gamma_m\left(\frac{\nu^{(i-1)}}{2}\right) \\
- \frac{1}{2} tr\left\{ S^{(i-1)}E_{q_i}(\Sigma^{-1}) \right\} - \frac{\nu^{(i-1)} + m + 1}{2} E_{q_i} \{ \log |\Sigma| \} \\
= \frac{\nu^{(i-1)}}{2} \log |S^{(i-1)}| - \frac{\nu^{(i-1)}m}{2} \log(2) - \log \Gamma_m\left(\frac{\nu^{(i-1)}}{2}\right) \\
- \frac{1}{2} tr\left\{ S^{(i-1)}(S^{(i)})^{-1}\nu^{(i)} \right\} \\
- \frac{\nu^{(i-1)} + m + 1}{2} \left\{ -\psi_m\left(\frac{\nu^{(i)}}{2}\right) - m \log(2) + \log |S^{(i)}| \right\},
\]

\[
E_{q_i} \{ \log(q_i(\Sigma)) \} = \frac{\nu^{(i)}}{2} \log |S^{(i)}| - \frac{\nu^{(i)}m}{2} \log(2) - \log \Gamma_m\left(\frac{\nu^{(i)}}{2}\right) - \frac{1}{2} tr\left\{ S^{(i)}E_{q_i}(\Sigma^{-1}) \right\} \\
= \frac{\nu^{(i)}}{2} \log |S^{(i)}| - \frac{\nu^{(i)}m}{2} \log(2) - \log \Gamma_m\left(\frac{\nu^{(i)}}{2}\right) - \frac{1}{2} tr\left\{ S^{(i)}(S^{(i)})^{-1}\nu^{(i)} \right\} \\
- \frac{\nu^{(i)} + m + 1}{2} \left\{ -\psi_m\left(\frac{\nu^{(i)}}{2}\right) - m \log(2) + \log |S^{(i)}| \right\}.
\]

Next, observe that if \( Z \) is matrix-variate normal distribution \( N_{s,t}(C, V \otimes W) \), then \( E(Z^TBZ) = Wtr(VB^T) + C^TBC \). Moreover, for \( A \) a \( r \times s \) matrix, we have
Therefore, we have

\[
E_q \{ \log(q_{i-1}(\beta_j)) \} = \frac{(N+1)m}{2} \log(2\pi) - \frac{N+1}{2} E_q(\log |\Sigma|) - \frac{m}{2} \log |(V_j^{(i-1)})^{-1}| - \frac{1}{2} \text{tr} \left[ \Sigma^{-1} (\beta_j - \hat{\beta}_j^{(i-1)})^T V_j^{(i-1)} (\beta_j - \hat{\beta}_j^{(i-1)}) \right]
\]

\[
= \frac{(N+1)m}{2} \log(2\pi) - \frac{N+1}{2} E_q(\log |\Sigma|) - \frac{m}{2} \log |(V_j^{(i-1)})^{-1}| - \frac{1}{2} \text{tr} \left[ \nu(i)(S^{(i)})^{-1} (\beta_j^{(i)} - \hat{\beta}_j^{(i-1)}) V_j^{(i-1)} (\beta_j^{(i)} - \hat{\beta}_j^{(i-1)}) \right]
+ E_q \left( \Sigma^{-1} \Sigma \text{tr}((V_j^{(i)})^{-1} V_j^{(i-1)}) \right) - \frac{m}{2} \log |(V_j^{(i)})^{-1}| - \frac{1}{2} m(N+1)
\]

where

\[
E_q \{ \log(y_i|\theta_j) \} = -\frac{m}{2} \log(2\pi) - \frac{m}{2} E_q(\log |\Sigma|) - \frac{1}{2} E_q(\log |\Sigma|) - \frac{1}{2} E_q \left( \frac{1}{\tau} \right) E_q \{ y_i^T \Sigma^{-1} y_i - 2 E_i^T \beta_j \Sigma^{-1} y_i + E_i^T \beta_j \Sigma^{-1} \beta_j^T E_i \}
\]

\[
= -\frac{m}{2} \log(2\pi) - \frac{m}{2} \left( \log(b^{(i)}_r) - \psi(a^{(i)}_r) \right) - \frac{1}{2} \left( -\psi_m \left( \frac{\nu(i)}{2} \right) - m \log(2) + \log |S^{(i)}| \right) - \frac{1}{2} \frac{\nu(i)}{b^{(i)}_r} \nu(i) y_i^T (S^{(i)})^{-1} y_i - 2 \nu(i) E_i^T \hat{\beta}_j^{(i)} (S^{(i)})^{-1} y_i + m E_i^T \Omega_j^{(i)} E_i + (E_i^T \hat{\beta}_j^{(i)})^T (\nu(i) (S^{(i)})^{-1}) (E_i^T \hat{\beta}_j^{(i)}) \}
\]
Finally, we have
\[
E_q_i \{ \log(q_i(\delta_i)) \} = \sum_{j=1}^{T} \hat{q}_i(\delta_i = j) \log(\hat{q}_i(\delta_i = j)).
\]

Acknowledgments. Victor Ong, David Nott and Ajay Jasra were supported by a Singapore Ministry of Education Tier 2 grant.

REFERENCES

[1] H. Attias, A variational Bayesian framework for graphical models, in Advances in Neural Information Processing Systems 12, MIT Press, 2000, 209–215.
[2] M. A. Beaumont, W. Zhang and D. J. Balding, Approximate Bayesian computation in population genetics, Genetics, 162 (2002), 2025–2035.
[3] D. M. Blei and M. I. Jordan, Variational inference for Dirichlet process mixtures, Bayesian Analysis, 1 (2006), 121–143.
[4] M. G. B. Blum, Approximate Bayesian computation: A nonparametric perspective, Journal of the American Statistical Association, 105 (2010), 1178–1187.
[5] M. G. B. Blum and O. François, Non-linear regression models for approximate Bayesian computation, Statistics and Computing, 20 (2010), 63–75.
[6] M. G. Blum and V. C. Tran, HIV with contact tracing: A case study in approximate Bayesian computation, Biostatistics, 11 (2010), 644–660.
[7] G. Box, Sampling and Bayes’ inference in scientific modelling and robustness (with discussion), Journal of the Royal Statistical Society, Series A, 143 (1980), 383–430.
[8] M. Bryant and E. B. Sudderth, Truly nonparametric online variational inference for hierarchical Dirichlet processes, in Advances in Neural Information Processing Systems 25, 2012, 2708–2716.
[9] M. Evans and G. H. Jang, Weak informativity and the information in one prior relative to another, Statist. Science, 26 (2011), 423–439.
[10] M. Evans and H. Moshonov, Checking for prior-data conflict, Bayesian Analysis, 1 (2006), 893–914.
[11] A. Gelman, Prior distributions for variance parameters in hierarchical models, Bayesian Analysis, 1 (2006), 515–533.
[12] A. Gelman, A. Jakulin, M. G. Pittau and Y.-S. Su, A weakly informative default prior distribution for logistic and other regression models, The Annals of Applied Statistics, 2 (2008), 1360–1383.
[13] A. K. Gupta and D. K. Nagar, Matrix Variate Distributions, Chapman & Hall/CRC Monographs and Surveys in Pure and Applied Mathematics, 104, Chapman & Hall/CRC, Boca Raton, FL, 2000.
[14] L. A. Hannah, D. M. Blei and W. B. Powell, Dirichlet process mixtures of generalized linear models, Journal of Machine Learning Research, 12 (2011), 1923–1953.
[15] M. D. Hoffman, D. M. Blei, C. Wang and J. Paisley, Stochastic variational inference, Journal of Machine Learning Research, 14 (2013), 1303–1347.
[16] M. C. Hughes and E. Sudderth, Memoized online variational inference for dirichlet process mixture models, in Advances in Neural Information Processing Systems 26 (eds. C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani and K. Q. Weinberger), Curran Associates, Inc., 2013, 1133–1141.
[17] V. Huynh and D. Phung, Streaming clustering with Bayesian nonparametric models, Special Issue on Machine Learning, Neurocomputing, 258 (2017), 52–62.
[18] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola and L. K. Saul, An introduction to variational methods for graphical models, Machine Learning, 37 (1999), 183–233.
[19] S. T. Kabisa, D. B. Dunson and J. S. Morris, Online variational Bayes inference for high-dimensional correlated data, Journal of Computational and Graphical Statistics, 25 (2016), 426–444.
[20] K. Kuribara, M. Welling and Y. W. Teh, Collapsed variational Dirichlet process mixture models, in Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI’07, 2007, 2796–2801.
[21] D. Lin, Online learning of nonparametric mixture models via sequential variational approximation, in Advances in Neural Information Processing Systems 26 (eds. C. Burges, L. Bottou, M. Welling, Z. Ghahramani and K. Weinberger), Curran Associates, Inc., 2013, 395–403.

[22] J. Luts, T. Broderick and M. P. Wand, Real-time semiparametric regression, Journal of Computational and Graphical Statistics, 23 (2014), 589–615.

[23] S. Montagna, S. T. Tokdar, Neelon and D. B. Dunson, Bayesian latent factor regression for functional and longitudinal data, Biometrics, 68 (2012), 1064–1073.

[24] D. J. Nott, C. C. Drovandi, K. Mengersen and M. Evans, Approximation of Bayesian predictive p-values with regression ABC, Bayesian Analysis, 13 (2018), 59–83.

[25] J. Ormerod and M. Wand, Explaining variational approximations, The American Statistician, 64 (2010), 140–153.

[26] A. Racine, A. P. Grieve, H. Flühler and A. F. M. Smith, Bayesian methods in practice: Experiences in the pharmaceutical industry, J. Roy. Statist. Soc. Ser. C, 35 (1986), 93–150.

[27] S. Ray and B. Mallick, Functional clustering by Bayesian wavelet methods, J. R. Stat. Soc. Ser. B Stat. Methodol., 68 (2006), 305–332.

[28] M.-A. Sato, Online model selection based on the variational Bayes, Neural Computation, 13 (2001), 1649–1681.

[29] B. Shahbaba and R. Neal, Nonlinear models using Dirichlet process mixtures, Journal of Machine Learning Research, 10 (2009), 1829–1850.

[30] A. Tank, N. Foti and E. Fox, Streaming Variational Inference for Bayesian Nonparametric Mixture Models, in Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics (eds. G. Lebanon and S. V. N. Vishwanathan), Proceedings of Machine Learning Research, 38, PMLR, San Diego, California, USA, 2015, 968–976.

[31] A. Tsanas and A. Xifara, Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools, Energy and Buildings, 49 (2012), 560–567.

[32] C. Wang and D. M. Blei, Truncation-free online variational inference for Bayesian nonparametric models, in Advances in Neural Information Processing Systems 25 (eds. F. Pereira, C. Burges, L. Bottou and K. Weinberger), Curran Associates, Inc., 2012, 413–421.

[33] C. Wang, J. Paisley and D. M. Blei, Online variational inference for the hierarchical Dirichlet process, in Proc. of the 14th Int’l. Conf. on Artificial Intelligence and Statistics (AISTATS), vol. 15, 2011, 752–760, URL http://jmlr.csail.mit.edu/proceedings/papers/v15/wang11a/wang11a.pdf.

[34] L. Wang and D. B. Dunson, Fast Bayesian inference in Dirichlet process mixture models, Journal of Computational and Graphical Statistics, 20 (2011), 196–216, Supplementary material available online.

[35] S. Waterhouse, D. MacKay and T. Robinson, Bayesian methods for mixture of experts, in Advances in Neural Information Processing Systems 8, MIT Press, 1996, 351–357.

[36] X. Zhang, D. J. Nott, C. Yau and A. Jasra, A sequential algorithm for fast fitting of Dirichlet process mixture models, Journal of Computational and Graphical Statistics, 23 (2014), 1143–1162.

[37] Z. Zhang, G. Dai and M. Jordan, Matrix-variate Dirichlet process mixture models, Proceedings of the Thirteenth Conference on Artificial Intelligence and Statistics (AISTATS), 9 (2010), 988–995.

[38] Z. Zhang, D. Wang, G. Dai and M. I. Jordan, Matrix-variate Dirichlet process priors with applications, Bayesian Analysis, 9 (2014), 259–286.

E-mail address: victor84@u.nus.edu
E-mail address: standj@nus.edu.sg
E-mail address: trchoi@korea.ac.kr
E-mail address: staja@nus.edu.sg