Variational Bayes for Hierarchical Mixture Models

Muting Wan*, James G. Booth†, Martin T. Wells ‡

Abstract

In recent years, sparse classification problems have emerged in many fields of study. Finite mixture models have been developed to facilitate Bayesian inference where parameter sparsity is substantial. Classification with finite mixture models is based on the posterior expectation of latent indicator variables. These quantities are typically estimated using the expectation-maximization (EM) algorithm in an empirical Bayes approach or Markov chain Monte Carlo (MCMC) in a fully Bayesian approach. MCMC is limited in applicability where high-dimensional data are involved because its sampling-based nature leads to slow computations and hard-to-monitor convergence. In this article, we investigate the feasibility and performance of variational Bayes (VB) approximation in a fully Bayesian framework. We apply the VB approach to fully Bayesian versions of several finite mixture models that have been proposed in bioinformatics, and find that it achieves desirable speed and accuracy in sparse classification with finite mixture models for high-dimensional data. Supplementary materials for this article that contain detailed technical derivations are available online.

Key Words: Bayesian inference; Markov chain Monte Carlo; Linear mixed models; Generalized linear mixed models; Bioinformatics.

*Muting Wan is a graduate student, Department of Statistical Science, Cornell University (email: mw545@cornell.edu).

†James Booth is Professor, Department of Biological Statistics and Computational Biology, Cornell University.

‡Martin Wells is Professor, Department of Statistical Science, Cornell University.

Professors Booth and Wells acknowledge the support of NSF-DMS 1208488.

1 Introduction

Variational Bayes (VB) methods in statistics arose from the family of variational approximation methods (Jaakkola, 2000) for performing approximate Bayesian inference for graphical models with latent variables (Bishop, 1999; Attias, 2000; Beal, 2003). Since then, VB has been promoted and employed in several fields of modern applications, such as signal processing (Smídl and Quinn, 2005; Blei and Jordan, 2006; Tzikas et al., 2008), political science (Grimmer, 2011), bioinformatics (Logsdon et al., 2010; Li and Sillanpää, 2012), and in medical research such as brain imaging (Friston et al., 2011; Goldsmith et al., 2011). In the early years, VB was applied to Gaussian mixture models, factor analysis, principal component analysis, hidden Markov models, and their mixtures, for model learning (Bishop, 1999; Ghahramani and Beal, 2000; Bishop et al., 2002; Beal, 2003) and model selection (Corduneanu and Bishop, 2001; Teschendorff et al., 2005; McGrory and Titterington, 2007). Since then, VB has also been used for learning nonlinear latent variable models (Honkela and Valpola, 2005; Salter-Townshend and Murphy, 2009), conducting functional regression analysis (Goldsmith et al., 2011), dealing with missing data in regression (Faes et al., 2011), and fitting location-scale models that contain elaborate distributional forms (Wand et al., 2011). Recently, Logsdon et al. (2010) applies VB based on the model proposed by Zhang et al. (2005) and shows that their VB solution outperforms single-marker testing in QTL analysis. Li et al. (2011) utilizes VB as an alternative to Markov chain Monte Carlo (MCMC) for hierarchical shrinkage-based regression models in QTL mapping with epistasis.

VB has been promoted as a fast deterministic method of approximating marginal posterior distributions and therefore as an alternative to MCMC methods for Bayesian inference (Beal, 2003; Bishop, 2006; Ormerod and Wand, 2010). Posterior means computed from VB approximated marginal posterior density have been observed to be extremely accurate. For example, Bishop (2006) illustrates the VB solution of a simple hierarchical Gaussian model. Comparison with Gelman et al. (2003), where the true marginal posteriors for the same model is provided, shows that the VB solution in Bishop (2006) recovers the posterior mean

exactly. However, it has also been observed that VB often underestimates posterior variance. This property is explained in Bishop (2006) to be due to the form of Kullback-Leibler divergence employed in the VB theory. In the setting where n observations follow a Gaussian distribution with large known variance and a zero-mean Gaussian-distributed mean whose precision is assigned a Gamma prior, Rue et al. (2009) uses a simple latent Gaussian model to illustrate that VB may underestimate posterior variance of the precision parameter by a ratio of $O(n)$. Ormerod (2011) proposes a grid-based method (GBVA) that corrects the variance in VB approximated marginal posterior densities for the same model.

Finite mixture models have been widely used for model-based classification (McLachlan and Peel, 2004; Zhang et al., 2005). Mixtures of distributions provide flexibility in capturing complex data distributions that cannot be well described by a single standard distribution. Finite mixture models, most commonly finite Gaussian mixture models, facilitate classification where each component distribution characterizes a class of "similar" data points.

Many problems in bioinformatics concern identifying sparse non-null features in a high-dimensional noisy null features space. This naturally leads to sparse classification with a finite mixture of "null" and "non-null" component distributions and latent indicator variables of component membership. The two-groups model in Efron (2008) is a finite mixture model with an implicit latent indicator variable for detection of non-null genes from a vast amount of null genes in microarray analysis. Bayesian classification results are obtained based on posterior expectation of the latent indicators. Typically, latent indicator variables are treated as missing data for an EM algorithm to be implemented in an empirical Bayes approach. Alternatively, fully Bayesian inference based on hierarchical structure of the model can be conducted via MCMC which approximates marginal posterior distributions. Finite-mixture-model-based classification allows strength to be borrowed across features in high-dimensional problems, in particular "large p small n" problems where p, the number of features to be classified, is several orders of magnitude greater than n, the sample size. However, in such high-dimensional problems it is difficult to assess convergence of fully Bayesian methods

implemented using MCMC algorithms and the computational burden may be prohibitive.

In the context of finite mixture models, model complexity prevents exact marginal posteriors from being derived explicitly. For example, consider the following simple two-component mixture model in a fully Bayesian framework, with observed data $\{d_g\}$ and known hyperparameters $\tau_0, \sigma^2_{\tau_0}, \psi_0, \sigma^2_{\psi_0}, a_0, b_0, \alpha_1, \alpha_0$:

$$\begin{aligned} d_g|b_g, \tau, \psi, \sigma^2 &= (1-b_g)N(\tau, \sigma^2) + b_g N(\tau+\psi, \sigma^2), \qquad (1) \\ b_g|p \sim Bernoulli(p), &\qquad \tau \sim N(\tau_0, \sigma^2_{\tau_0}), \qquad \psi \sim N(\psi_0, \sigma^2_{\psi_0}), \\ \sigma^2 \sim IG(a_0, b_0), &\qquad p \sim Beta(\alpha_1, \alpha_0), \end{aligned}$$

where $IG(a, b)$ denotes an Inverse-Gamma distribution with shape a and scale b. Classification is conducted based on the magnitude of the posterior mean of the latent indicator b_g; a posterior mean close to 1 indicating membership in the non-null group for feature g. The Integrated Nested Laplace Approximations (INLA) framework is another popular approximate Bayesian method, introduced in Rue et al. (2009), whose implementation is available in the `R-INLA` package (Martino and Rue, 2009). However, INLA does not fit any of the mixture models mentioned above because they are not members of the class of latent Gaussian models. In the context of model (1), numerical experiments show that MCMC produces believable approximate marginal posterior densities which can be regarded as proxies of the true marginal posteriors despite the label-switching phenomenon (Marin and Robert, 2007) well-known for finite mixture models. In Section 2, we show that VB approximated densities are comparable to MCMC approximated ones in terms of both posterior mean and variance in this context, but VB achieves substantial gains in computational speed over MCMC.

More general finite mixture models are common for (empirical and fully) Bayesian inference in high-dimensional application areas, such as bioinformatics. Computations for the two-groups model in Smyth (2004), for example, are burdensome due to presence of the gene-specific error variance and different variances for the two component distributions. Fully Bayesian inference via MCMC procedures are limited in practicality due to heavy computational burden resulting from the high-dimensionality of the data.

Our objective in this article is to investigate the feasibility and performance of VB for finite mixture models in sparse classification in a fully Bayes framework. We will see in later sections that there are significant computational issues with MCMC implementations in this context, making this approach impractical. In contrast, VB is fast, accurate, and easy to implement due to its deterministic nature. Moreover, the VB algorithm results in a very accurate classifier, despite the fact that it significantly underestimates the posterior variances of model parameters in some cases. To our knowledge, GBVA has not been evaluated before in this setting and our investigation indicates that it does not result in improved accuracy while adding substantially to the computational burden.

The plan for this article is as follows. Section 2 reviews theory underlying the VB method and approximation of marginal posterior densities via VB for the aforementioned simple two-component mixture model. Motivation behind using a hierarchical mixture model framework and implementation of VB for general finite mixture models are outlined in Section 3. We illustrate application of VB via examples involving simulated and real data in Section 4. Finally, we conclude with some discussion in Section 5. Many of the technical arguments are given in the supplementary materials.

2 Variational Bayes

2.1 Overview of the VB method

VB is a deterministic estimation methodology based on a factorization assumption on the approximate joint posterior distribution. This is a free-form approximation in that implementation of VB does not start with any assumed parametric form of the posterior distributions. The free-form factors are approximate marginal posterior distributions that one tries to obtain according to minimization of the Kullback-Leibler divergence between the approximate joint posterior and the true joint posterior. As a tractable solution to the minimization problem, the optimal factors depend on each other, which naturally leads to an iterative

scheme that cycles through an update on each factor. The convergence of the algorithm is monitored via a single scalar criterion. Ormerod and Wand (2010) and Chapter 10 of Bishop (2006) present detailed introduction to VB in statistical terms. The VB approximation in this article refers to variational approximation to the joint and marginal posterior distributions under, in terminology used in Ormerod and Wand (2010), the product density (independence) restriction. The concept of a mean-field approximation has its roots in statistical physics and is a form of VB approximation under a more stringent product density restriction that the approximate joint posterior fully factorizes. The basic VB theory is outlined as follows.

Consider a Bayesian model with observed data $\mathbf{y}$, latent variables $\mathbf{x}$ and parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_m\}$. Denote $H = \{\mathbf{x}, \boldsymbol{\Theta}\}$ as the collection of all unknown quantities. The posterior density $p(H|\mathbf{y}) = p(\mathbf{y}, H)/p(\mathbf{y})$ is not necessarily tractable due to the integral involved in computing the marginal data density $p(\mathbf{y})$. This intractability prevents computing marginal posterior densities such as $p(\boldsymbol{\theta}_i|\mathbf{y}) = \int p(H|\mathbf{y}) d\{-\boldsymbol{\theta}_i\}$ where $\{-\boldsymbol{\theta}_i\}$ refers to $H \backslash \{\boldsymbol{\theta}_i\} = \{\mathbf{x}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_{i-1}, \boldsymbol{\theta}_{i+1}, ..., \boldsymbol{\theta}_m\}$. Two key elements, namely minimization of the Kullback-Leibler divergence and employment of the product density restriction, underpin the VB algorithm. The Kullback-Leibler divergence gives rise to a scalar convergence criterion that governs the iterative algorithm. Through the product density restriction one assumes independence which allows derivation of tractable density functions.

For an arbitrary density for H, $q(H)$,

$$\log p(\mathbf{y}) = \int q(H) \log \left(\frac{p(\mathbf{y}, H)}{q(H)} \right) dH + \int q(H) \log \left(\frac{q(H)}{p(H|\mathbf{y})} \right) dH. \tag{2}$$

The second integral on the right hand side of (2) is the Kullback-Leibler divergence between $q(H)$ and $p(H|\mathbf{y})$, i.e. $D_{KL}(q(H) \| p(H|\mathbf{y}))$ which is non negative. Hence,

$$\log p(\mathbf{y}) \geq \int q(H) \log \left(\frac{p(\mathbf{y}, H)}{q(H)} \right) dH = C_q(\mathbf{y}),$$

where $C_q(\mathbf{y})$ denotes the lower bound on the log of the marginal data density. This lower bound depends on the density q. Given a data set, $\log p(\mathbf{y})$ is a fixed quantity. This means that if a density q is sought to minimize $D_{KL}(q(H) \| p(H|\mathbf{y}))$, q also maximizes $C_q(\mathbf{y})$ making

the lower bound a viable approximation to $\log p(\mathbf{y})$.

The product density restriction states that $q(H)$ factorizes into some partition of H, i.e. $q(H) = \prod_{i=1}^{k} q_i(\mathbf{h}_i)$, but the parametric form of the $q_i(\mathbf{h}_i)$ factors is not specified. It can be shown that, under the product density restriction, the optimal $q_i(\mathbf{h}_i)$ takes the following form (Ormerod and Wand, 2010):

$$q_i(\mathbf{h}_i) \propto \exp\{E_{-\mathbf{h}_i}(\log p(\mathbf{y}, H))\} \tag{3}$$

for $i = 1, .., k$, where expectation is taken over all unknown quantities except $\mathbf{h}_i$ with respect to the density $\prod_{j \neq i}^{k} q_j(\mathbf{h}_j)$.

Since each variational posterior $q_i(\mathbf{h}_i)$ depends on other variational posterior quantities, the algorithm involves iteratively updating $q_i(\mathbf{h}_i), 1 \leq i \leq k$ until the increase in $C_q(\mathbf{y})$, computed in every iteration after each of the updates $q_i(\mathbf{h}_i), 1 \leq i \leq k$ has been made, is negligible. Upon convergence, approximate marginal posterior densities ${q^*}_i(\mathbf{h}_i), 1 \leq i \leq k$ are obtained, as well as an estimate $C_{q^*}(\mathbf{y})$ of the log marginal data density $\log p(\mathbf{y})$.

2.2 Practicality

As noted in Faes et al. (2011), under mild assumptions convergence of the VB algorithm is guaranteed (Luenberger and Ye, 2008, pg. 253). Bishop (2006) refers to Boyd and Vandenberghe (2004) and states that convergence is guaranteed because the lower bound $C_q(\mathbf{y})$ is convex with respect to each of the factors $q_i(\mathbf{h}_i)$. Convergence is easily monitored through the scalar quantity $C_q(\mathbf{y})$. Moreover, upon convergence, since $C_{q^*}(\mathbf{y})$ approximates $\log p(\mathbf{y})$, VB can be used to compare solutions produced under the same model but with different starting values or different orders of updating the q-densities. Because the optimal set of starting values corresponds to the largest $C_{q^*}(\mathbf{y})$ upon convergence, starting values can be determined empirically with multiple runs of the same VB algorithm. Experiments on starting values are generally undemanding due to high computational efficiency of the VB method, and are especially useful for problems where the true solutions are multimodal.

Computational convenience of VB is achieved if conjugate priors are assigned for model

parameters and the complete-data distribution belongs to the exponential family. Beal (2003) describes models that satisfy these conditions as conjugate-exponential models. In these models the approximate marginal posterior densities $q_i(\mathbf{h}_i)$ take the conjugate form and the VB algorithm only requires finding the characterizing parameters. The induced conjugacy plays an important role in producing analytical solutions. Feasibility of VB for models with insufficient conjugacy is an open research area.

In practice, it suffices to impose some relaxed factorization of $q(H)$ as long as the factorization allows derivation of a tractable solution. Although, in many cases, the chosen product density restriction leads to an induced product form of factors of $q(H)$, the imposed factorization and induced factorization of $q(H)$ are generally not equivalent and may lead to different posterior independence structures.

As an introductory example, consider a simple fully Bayesian model with a $N(\mu, \sigma^2)$ data distribution and a bivariate Normal-Inverse-Gamma prior $p_{(\mu,\sigma^2)}(\mu, \sigma^2) = p_{\mu|\sigma^2}(\mu|\sigma^2)p_{\sigma^2}(\sigma^2)$ with $p_{\mu|\sigma^2}(\mu|\sigma^2) = N(\mu_0, \lambda_0\sigma^2)$ and $p_{\sigma^2}(\sigma^2) = IG(A_0, B_0)$. Imposing the product density restriction $q_{(\mu,\sigma^2)}(\mu, \sigma^2) = q_\mu(\mu)q_{\sigma^2}(\sigma^2)$ leads to independent Normal marginal posterior $q_\mu(\mu)$ and Inverse-Gamma marginal posterior $q_{\sigma^2}(\sigma^2)$. However, not imposing the product density restriction in this case leads to the bivariate Normal-Inverse-Gamma joint posterior $q_{\mu|\sigma^2}(\mu|\sigma^2)q_{\sigma^2}(\sigma^2)$ that reflects a different posterior independence structure. Hence, although choice of prior and product density restriction is problem-based, caution should be taken in examining possible correlation between model parameters, because imposing too much factorization than needed risks poor VB approximations if dependencies between the hidden quantities are necessary. This type of degradation in VB accuracy is noted in Ormerod and Wand (2010). There is a trade-off between tractability and accuracy with any type of problem simplification via imposed restrictions.

2.3 Over-confidence

Underestimation of posterior variance of the VB approximate density has been observed in different settings such as those in Wang and Titterington (2005), Consonni and Marin (2007), and Rue et al. (2009). We review a possible reason why the variance of the approximate posterior given by VB tends to be smaller than that of the true posterior.

As explained previously by (2), a density $q(H)$ is sought to minimize the Kullback-Leibler divergence between itself and the true joint posterior $p(H|\mathbf{y})$. Upon examination of the form of the Kullback-Leibler divergence used in the VB method,

$$D_{KL}(q(H)\|p(H|\mathbf{y})) = \int q(H)\left(-\log\left(\frac{p(H|\mathbf{y})}{q(H)}\right)\right)dH,$$

Bishop (2006) points out that, in regions of H space where density $p(H|\mathbf{y})$ is close to zero, $q(H)$, the minimizer, has to be close to zero also; otherwise the negative log term in the integrand would result in a large positive contribution to the Kullback-Leibler divergence. Thus, the optimal $q^*(H)$ tends to avoid regions where $p(H|\mathbf{y})$ is small. The factorized form of $q^*(H)$ implies that each of the factor q^*-densities in turn tends to avoid intervals where the true marginal posterior density is small. Thus, the marginal posterior densities approximated by VB are likely to be associated with underestimated posterior variance and a more compact shape than the true marginal posterior densities.

2.4 Simple two-component mixture model

For model (1) where $g = 1, 2, ..., G$, with known hyperparameters $\tau_0, \sigma^2_{\tau_0}, \psi_0, \sigma^2_{\psi_0}, a_0, b_0, \alpha_1, \alpha_0$, observed data $E = \{d_g\}$ and hidden quantities $H = \{\tau, \psi, \sigma^2, p, \{b_g\}\}$, imposing the product density restriction $q(H) = q_{\{b_g\}}(\{b_g\}) \times q_{(\tau,p)}(\tau, p) \times q_\psi(\psi) \times q_{\sigma^2}(\sigma^2)$ results in q-densities $q_{\{b_g\}}(\{b_g\}) = \prod_g q_{b_g}(b_g) = \prod_g Bernoulli\left(\frac{\exp(\hat{\rho}_{1g})}{\exp(\hat{\rho}_{1g}) + \exp(\hat{\rho}_{0g})}\right)$, $q_\tau(\tau) = N(\widehat{M_\tau}, \widehat{V_\tau})$, $q_p(p) = Beta(\hat{\alpha}_1, \hat{\alpha}_0)$, $q_\psi(\psi) = N(\widehat{M_\psi}, \widehat{V_\psi})$, and $q_{\sigma^2}(\sigma^2) = IG(\hat{a}, \hat{b})$. Optimal q-densities are found by employing the following iterative scheme for computing variational parameters.

i) Set $\hat{a} = \frac{G}{2} + a_0$. Initialize $\widehat{M_{\sigma^{-2}}} = 1$, $\hat{\rho}_{0g} = \hat{\rho}_{1g} = 0$ for each g, and

$$\widehat{M_{b_g}} = \begin{cases} 1 & \text{if } \operatorname{rank}(d_g) \geq 0.9G \\ 0 & \text{otherwise} \end{cases} \quad \text{for each } g,$$

$$\widehat{M_\psi} = \left| \sum_{g=1}^{G} d_g - \sum_{\{g:\operatorname{rank}(d_g)\geq 0.9G\}} d_g \right|.$$

ii) Update

$$\widehat{M_\tau} \leftarrow \frac{\sum_g \widehat{M_{\sigma^{-2}}} \left((1 - \widehat{M_{b_g}}) d_g + \widehat{M_{b_g}} (d_g - \widehat{M_\psi}) \right)}{G\widehat{M_{\sigma^{-2}}} + \frac{1}{\sigma^2_{\tau_0}}}$$

$$\widehat{M_\psi} \leftarrow \frac{\sum_g \widehat{M_{\sigma^{-2}}} \widehat{M_{b_g}} (d_g - \widehat{M_\tau})}{\sum_g \widehat{M_{b_g}} \widehat{M_{\sigma^{-2}}} + \frac{1}{\sigma^2_{\psi_0}}}$$

$$\hat{b} \leftarrow \frac{1}{2} \sum_g \left(\frac{1}{G\widehat{M_{\sigma^{-2}}} + \frac{1}{\sigma^2_{\tau_0}}} + \frac{\widehat{M_{b_g}}}{\sum_g \widehat{M_{b_g}} \widehat{M_{\sigma^{-2}}} + \frac{1}{\sigma^2_{\psi_0}}} \right)$$
$$+ \frac{1}{2} \sum_g \left((1 - \widehat{M_{b_g}})(d_g - \widehat{M_\tau})^2 + \widehat{M_{b_g}} (d_g - \widehat{M_\tau} - \widehat{M_\psi})^2 \right) + b_0$$

$$\widehat{M_{\sigma^{-2}}} \leftarrow \frac{\hat{a}}{\hat{b}}$$

$$\hat{\rho}_{1g} \leftarrow \frac{-\widehat{M_{\sigma^{-2}}}}{2} \left(\frac{1}{G\widehat{M_{\sigma^{-2}}} + \frac{1}{\sigma^2_{\tau_0}}} + \frac{1}{\sum_g \widehat{M_{b_g}} \widehat{M_{\sigma^{-2}}} + \frac{1}{\sigma^2_{\psi_0}}} + (d_g - \widehat{M_\tau} - \widehat{M_\psi})^2 \right) + \widehat{\log p}$$

$$\hat{\rho}_{0g} \leftarrow \frac{-\widehat{M_{\sigma^{-2}}}}{2} \left(\frac{1}{G\widehat{M_{\sigma^{-2}}} + \frac{1}{\sigma^2_{\tau_0}}} + (d_g - \widehat{M_\tau})^2 \right) + \widehat{\log(1-p)}$$

$$\widehat{M_{b_g}} \leftarrow \frac{\exp(\hat{\rho}_{1g})}{\exp(\hat{\rho}_{1g}) + \exp(\hat{\rho}_{0g})}.$$

iii) Repeat ii) until the increase in

$$C_q(\mathbf{y}) = \sum_g \left\{ -\frac{1}{2} \log(2\pi) - \frac{\widehat{M_{\sigma^{-2}}}}{2} \times (1 - \widehat{M_{b_g}}) \left(\frac{1}{G\widehat{M_{\sigma^{-2}}} + \frac{1}{\sigma^2_{\tau_0}}} + (d_g - \widehat{M_\tau})^2 \right) \right.$$
$$\left. - \frac{\widehat{M_{\sigma^{-2}}}}{2} \times \widehat{M_{b_g}} \left(\frac{1}{G\widehat{M_{\sigma^{-2}}} + \frac{1}{\sigma^2_{\tau_0}}} + \frac{1}{\sum_g \widehat{M_{b_g}} \widehat{M_{\sigma^{-2}}} + \frac{1}{\sigma^2_{\psi_0}}} + (d_g - \widehat{M_\tau} - \widehat{M_\psi})^2 \right) \right\}$$
$$- \sum_g \left\{ \widehat{M_{b_g}} \log \widehat{M_{b_g}} + (1 - \widehat{M_{b_g}}) \log(1 - \widehat{M_{b_g}}) \right\}$$

$$+\log\left(Beta\left(\sum_g \widehat{M_{b_g}}+\alpha_1,\sum_g(1-\widehat{M_{b_g}})+\alpha_0\right)\right)-\log\left(Beta(\alpha_1,\alpha_0)\right)$$

$$+\frac{1}{2}\left(\log\left(\frac{1}{G\widehat{M_{\sigma^{-2}}}+\frac{1}{\sigma^2_{\tau_0}}}\right)-\log\sigma^2_{\tau_0}+1-\frac{\frac{1}{G\widehat{M_{\sigma^{-2}}}+\frac{1}{\sigma^2_{\tau_0}}}+\left(\widehat{M_\tau}-\tau_0\right)^2}{\sigma^2_{\tau_0}}\right)$$

$$+\frac{1}{2}\left(\log\left(\frac{1}{\sum_g \widehat{M_{b_g}}\widehat{M_{\sigma^{-2}}}+\frac{1}{\sigma^2_{\psi_0}}}\right)-\log\sigma^2_{\psi_0}+1\right.$$

$$\left.-\frac{\frac{1}{\sum_g \widehat{M_{b_g}}\widehat{M_{\sigma^{-2}}}+\frac{1}{\sigma^2_{\psi_0}}}+\left(\widehat{M_\psi}-\psi_0\right)^2}{\sigma^2_{\psi_0}}\right)$$

$$+a_0\log b_0-\log\Gamma(a_0)-\frac{b_0\hat{a}}{\hat{b}}-\hat{a}\log(\hat{b})+\log\Gamma(\hat{a})+\hat{a}$$

from previous iteration becomes negligible.

iv) Upon convergence, the remaining variational parameters are computed:

$$\widehat{V_\tau}\leftarrow\frac{1}{G\widehat{M_{\sigma^{-2}}}+\frac{1}{\sigma^2_{\tau_0}}}\qquad\text{and}\qquad\widehat{V_\psi}\leftarrow\frac{1}{\sum_g \widehat{M_{b_g}}\widehat{M_{\sigma^{-2}}}+\frac{1}{\sigma^2_{\psi_0}}}.$$

Here, we set starting values for $\widehat{M_{b_g}}$ such that $\widehat{M_{b_g}}=1$ for genes that correspond to the top 10% of d_g values and $\widehat{M_{b_g}}=0$ otherwise. Classification with a different set of starting values such that $\widehat{M_{b_g}}=1$ for genes that correspond to the top 5% and bottom 5% of d_g values and $\widehat{M_{b_g}}=0$ otherwise leads to almost the same results. In practice, we recommend using the "top-5%-bottom-5%" scheme of setting starting values for $\widehat{M_{b_g}}$ in the VB algorithm for classification with little prior information on location of the mixture component distributions rather than randomly generating values for $\widehat{M_{b_g}}$ - empirical evidence suggests that VB with randomly generated labels does not produce reasonable solutions.

2.5 Marginal posterior approximation

To investigate we simulated 20,000 values from the simple two-component model (1) with true values $\tau = 0$, $\psi = 20$, $p = 0.2$, $b_g \sim Bernoulli(p)$ *i.i.d.* for each g, and $\sigma^2 = 36$. In the Bayesian analysis τ and ψ were assigned $N(0, 100)$ priors, σ^2 an $IG(0.1, 0.1)$ prior, and p a $Beta(0.1, 0.9)$ prior. Starting values and priors used in VB, MCMC, and the base VB of GBVA are kept the same for comparison. MCMC was implemented in WinBUGS via R with 1 chain of length 20,000, a burn-in 15,000, and a thinning factor of 10. Figure 1 shows that VB-approximated marginal posterior densities closely match the MCMC-approximates. In particular, the posterior mean of $\{b_g\}$ estimated by VB, when plotted against posterior mean of $\{b_g\}$ estimated by MCMC, almost coincide with the intercept zero slope one reference line, indicating little difference between VB and MCMC in terms of classification. In fact, with 3972 simulated non-null genes and a 0.8 cutoff for classification, MCMC identified 3303 genes as non-null with true positive rate 0.803 and false positive rate 0.00705, and VB also detected 3291 genes as non-null with true positive rate 0.800 and the same false positive rate as MCMC. Moreover, on an Intel Core i5-2430M 2.40GHz, 6GB RAM computer, it took VB 1.26 seconds to reach convergence with a 10^{-6} error tolerance, whereas MCMC procedure took about 13 minutes. Both methods are suitable for classification and posterior inference, with VB demonstrating an advantage in speed, in this example.

The grid-based variational approximations (GBVA) method detailed in Ormerod (2011) was developed to correct for over-confidence in VB densities. The method involves running a base VB algorithm and subsequent VB algorithms for density approximation with the aid of numerical interpolation and integration. For each marginal posterior density of interest, re-applying VB over a grid of values in GBVA is similar in spirit to re-applying Laplace approximation in the INLA (Rue et al., 2009) method, and is the key step towards refining the marginal posterior density approximated by the base VB algorithm. We observe in Figure 1 that, although GBVA appears to correct for underestimation of posterior variance of VB, loss in accuracy occurs in estimation of posterior mean in comparison with MCMC. Experiments

with grid selection and precision of numerical methods did not correct the shift in posterior mean, and a systematic error caused by grid-based calculations of the unnormalized density is suspected. Although the shifts in posterior mean of GBVA densities observed in Figure 1 are small in scale, for posterior inference the mismatch in mean may incur more serious loss in accuracy than underestimation of posterior variance of VB densities. Moreover, in the example, the total run time of GBVA (almost 6 hours) is even larger than that of MCMC (13 minutes). The undesirable speed of GBVA is due to the fact that refining the marginal posterior of the latent variable $\{b_g\}$ must go through two iterations, i.e. two grid points 0 and 1, for each one of the G genes. Therefore, in GBVA implementation, this step alone requires $2 \times G$ iterations. This example suggests that for a "large p small n" problem, the approach of calculation over grid values can easily render the GBVA method computationally infeasible.

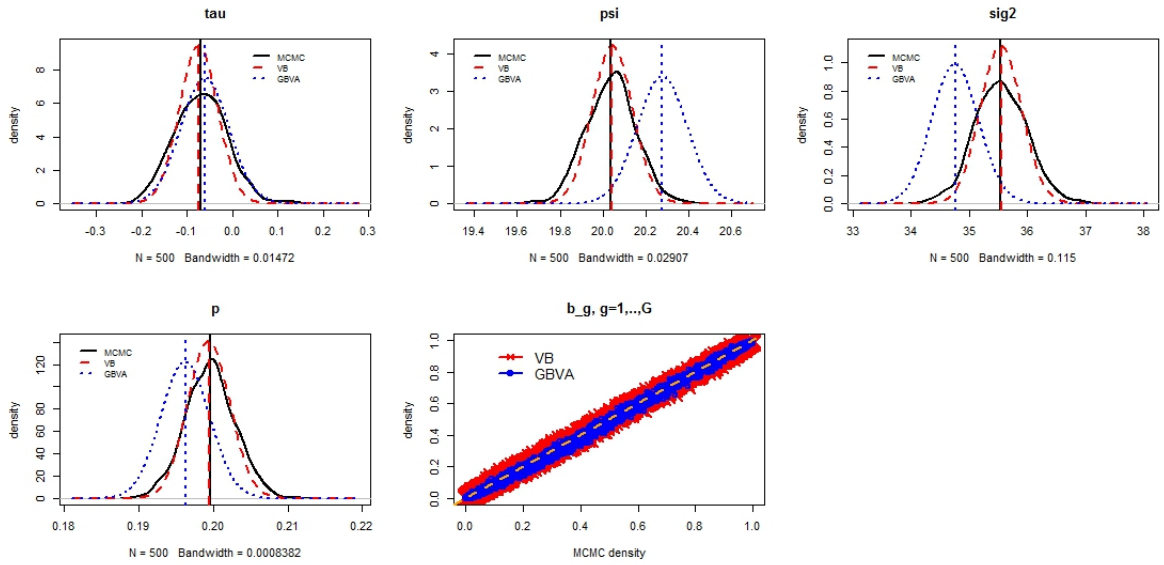

Figure 1: Approximate marginal posteriors given by MCMC, VB and GBVA for simulated data under the simple two-component mixture model (1). In the last plot, the posterior mean of b_g, $g = 1, .., G$ is compared between VB and MCMC, and between GBVA and MCMC.

3 VB for a general finite mixture model

3.1 Motivation

For large-scale problems that involve complex models on high-dimensional data, it is natural to consider MCMC for approximation of notoriously intractable marginal posterior densities. However, in many cases implementation of MCMC is impractical. MCMC, because of its sampling-based nature, becomes cumbersome for hierarchical "large p small n" models where high-dimensional unknown quantities on various hierarchies are inter-dependent. To achieve desirable accuracy of MCMC therefore requires (multiple) long chains. Yet convergence of high-dimensional MCMC chains is difficult to monitor (Cowles and Carlin, 1996), and optimal MCMC chain settings often have to be determined based on heuristics. Moreover, in mixture models, an MCMC chain may produce evaluation of a marginal posterior density that is the same as other chains but associated with different component labels. The label-switching issue has been noted, for example in De Freitas et al. (2001), Marin and Robert (2007), and Grimmer (2011). Without correction of label-switching, misleading classification may arise in cases where multiple chains are used and the MCMC solution, taken as an average over chains with different or even contrasting labels, is wrong.

A common strategy for improving efficiency in approximation problems is to make simplifying assumptions so that plausible solutions for the complicated problem are obtained with satisfactory speed and accuracy. This is the motivation behind implementing VB for approximation of marginal posterior densities for finite mixture models in sparse classification problems. In this context, VB is promising in achieving comparable accuracy to MCMC given limited computational resources, with easy-to-monitor convergence and no label-switching due to its deterministic nature, in scenarios where MCMC implementation is unattractive.

3.2 The B-LIMMA model

We illustrate a VB implementation for a general finite mixture models using a fully Bayesian version of the LIMMA model introduced by Smyth (2004).

The LIMMA model is designed for testing significance of differential gene expression in microarray experiments via individual moderated t-tests and posterior log odds ratios. Model parameters can be estimated via an empirical Bayes approach, from which closed form test-statistics can be obtained. For a two-sample experimental design, let y_{ijg} denote the normalized log expression of gene g in sample j from treatment group i, where $g = 1, ..., G$, $i = 1, 2$, and $j = 1, ..., n_{ig}$. Then $d_g = \bar{y}_{2.g} - \bar{y}_{1.g}$ represents the observed differential expression for gene g, and $m_g = \sum_i \sum_j^{n_{ig}} (y_{ijg} - \bar{y}_{i.g})^2 / f_g$, where $f_g = n_{1g} + n_{2g} - 2$, is the mean squared error. A fully Bayesian version of the LIMMA model in this context is the following hierarchical mixture model:

$$
\begin{aligned}
& d_g | b_g, \psi_g, \sigma^2_{\epsilon,g}, \tau = \tau + b_g \psi_g + \epsilon_g, \qquad (4) \\
& m_g | \sigma^2_{\epsilon,g} \sim \frac{\sigma^2_{\epsilon,g} \chi^2_{f_g}}{f_g}, \quad \text{where } f_g := n_{1g} + n_{2g} - 2, \\
& \psi_g | \nu, \sigma^2_{\epsilon,g} \sim N(0, \nu \sigma^2_{\epsilon,g}), \\
& \epsilon_g | \sigma^2_{\epsilon,g} \sim N(0, \sigma^2_g), \quad \text{where } \sigma^2_g := \sigma^2_{\epsilon,g} c_g, \quad c_g := \frac{1}{n_{1g}} + \frac{1}{n_{2g}}, \\
& b_g | p \sim Bernoulli(p) \quad i.i.d., \qquad \sigma^2_{\epsilon,g} \sim IG\left(A_\epsilon, B_\epsilon\right) \; i.i.d., \\
& \tau \sim N\left(\mu_{\tau_0}, \sigma^2_{\tau_0}\right), \qquad \nu \sim IG\left(A_\nu, B_\nu\right), \qquad p \sim Bernoulli\left(\alpha_1, \alpha_0\right).
\end{aligned}
$$

In this model τ represents the overall mean treatment difference, ψ_g is the gene-specific effect of the treatment, and b_g is a latent indicator which takes the value 1 if gene g is non-null, i.e. is differentially expressed in the two treatment groups.

We refer to the above model as the B-LIMMA model. Derivation of the VB algorithm for the B-LIMMA model, which we refer to as VB-LIMMA, is as follows.

The set of observed data and the set of unobserved data are identified as $\mathbf{y}$ and H, respectively. For the B-LIMMA model, $\mathbf{y} = \{\{d_g\}, \{m_g\}\}$ and $H = \{\{b_g\}, \{\psi_g\}, \{\sigma^2_g\}, \tau, \nu, p\}$.

The complete data log likelihood is

$$\log p(\mathbf{y}, H) = \sum_g \log p(d_g|b_g, \psi_g, \sigma_g^2, \tau) + \sum_g \log p(m_g|\sigma_g^2) + \sum_g \log p(\psi_g|\nu, \sigma_g^2)$$
$$+ \sum_g \log p(b_g|p) + \log p(\tau) + \log p(\nu) + \sum_g \log p(\sigma_g^2) + \log p(p).$$

In particular, the observed data log likelihood involves

$$\log p(d_g|b_g, \psi_g, \sigma_g^2, \tau) = \frac{1}{2}\log\left(2\pi\sigma_g^{-2}\right) - \frac{1}{2\sigma_g^2}\left[(1-b_g)(d_g-\tau)^2 + b_g(d_g-\tau-\psi_g)^2\right],$$

$$\log p(m_g|\sigma_g^2) = -\log\left(\frac{\sigma_g^2}{c_g f_g}\right) - \log\left(2^{\frac{f_g}{2}}\Gamma\left(\frac{f_g}{2}\right)\right)$$
$$+ \left(\frac{f_g}{2}-1\right)\log\left(\frac{m_g c_g f_g}{\sigma_g^2}\right) - \frac{1}{2}\left(\frac{m_g c_g f_g}{\sigma_g^2}\right).$$

The product density restriction is imposed such that $q(H) = q_{\{b_g\}}(\{b_g\}) \times q_{\{\psi_g\}}(\{\psi_g\}) \times q_{\{\sigma_g^2\}}(\{\sigma_g^2\}) \times q_{(\tau,\nu,p)}(\tau,\nu,p)$. Further factorizations are induced by applying (3):

$$q_{\{b_g\}}(\{b_g\}) = \prod_g q_{b_g}(b_g), \qquad q_{\{\psi_g\}}(\{\psi_g\}) = \prod_g q_{\psi_g}(\psi_g),$$

$$q_{\{\sigma_g^2\}}(\{\sigma_g^2\}) = \prod_g q_{\sigma_g^2}(\sigma_g^2), \quad \text{and} \quad q_{(\tau,\nu,p)}(\tau,\nu,p) = q_\tau(\tau)q_\nu(\nu)q_p(p).$$

Then, approximate marginal posterior distributions, with $\widehat{M_\theta}$ denoting the variational posterior mean $\int \theta q_\theta(\theta)d\theta$, and $\widehat{V_\theta}$ denoting the variational posterior variance $\int(\theta - \widehat{M_\theta})^2 q_\theta(\theta)d\theta$, are $q_\tau(\tau) = N(\widehat{M_\tau}, \widehat{V_\tau})$, $q_\nu(\nu) = IG(\widehat{A_\nu}, \widehat{B_\nu})$, $q_p(p) = Beta(\widehat{\alpha_1}, \widehat{\alpha_0})$, $q_{\psi_g}(\psi_g) = N(\widehat{M_{\psi_g}}, \widehat{V_{\psi_g}})$, $q_{\sigma_g^2}(\sigma_g^2) = IG(\widehat{A_{\sigma_g^2}}, \widehat{B_{\sigma_g^2}})$, and $q_{b_g}(b_g) = Bernoulli(\widehat{M_{b_g}})$. Updating the q-densities in an iterative scheme boils down to updating the variational parameters in the scheme. Convergence is monitored via the scalar quantity $C_q(\mathbf{y})$, the lower bound on the log of the marginal data density

$$C_q(\mathbf{y}) = E_{q(H)}\left\{\sum_g \log p(d_g|b_g, \psi_g, \sigma_g^2, \tau) + \sum_g \log p(m_g|\sigma_g^2)\right.$$
$$+ \sum_g \log p(\sigma_g^2) - \sum_g \log q(\sigma_g^2) + \log p(\tau) - \log q(\tau)$$
$$+ \sum_g \log p(\psi_g|\nu, \sigma_g^2) + \log p(\nu) - \sum_g \log q(\psi_g) - \log q(\nu)$$
$$\left. + \sum_g \log p(b_g|p) + \log p(p) - \sum_g \log q(b_g) - \log q(p)\right\}.$$

It is only necessary to update the variational posterior means $\widehat{M_\cdot}$ in the iterative scheme.

Upon convergence, the other variational parameters are computed based on the converged value of those involved in the iterations. Therefore, the VB-LIMMA algorithm consists of the following steps:

Step 1: Initialize $\widehat{A_\nu}, \widehat{B_\nu}, \{\widehat{A_{\sigma_g^2}}\}, \{\widehat{B_{\sigma_g^2}}\}, \{\widehat{M_{\psi_g}}\}, \{\widehat{M_{b_g}}\}$.

Step 2: Cycle through $\widehat{M_\tau}, \widehat{B_\nu}, \{\widehat{B_{\sigma_g^2}}\}, \{\widehat{M_{\psi_g}}\}, \{\widehat{M_{b_g}}\}$ iteratively, until the increase in $C_q(\mathbf{y})$ computed at the end of each iteration is negligible.

Step 3: Compute $\widehat{V_\tau}, \{\widehat{V_{\psi_g}}\}, \widehat{\alpha_1}, \widehat{\alpha_0}$ using converged variational parameter values.

4 Numerical Illustrations

4.1 Simulation

The goal of this simulation study is to assess the performance of VB as a classifier when various model assumptions are correct, and to determine the accuracy of VB approximation of marginal posterior distributions.

4.1.1 The B-LIMMA model

Under the B-LIMMA model (4), the difference in sample means $\{d_g\}$ and mean squared errors $\{m_g\}$ are simulated according to model (1) of Bar et al. (2010) with $\mu = 0$ and $\gamma_g \equiv 0$. VB-LIMMA approximates a marginal posterior density whose mean is then used as posterior estimate of the corresponding parameter for classification. Performance comparison is made between VB implemented in R and the widely-used `limma` (Smyth, 2005) package.

A data set containing $G = 5000$ genes was simulated under the B-LIMMA model with $n_{1g} \equiv n_{2g} \equiv 8, \tau = 0, \sigma_g^2 \sim IG(41, 400)$ *i.i.d.*, and ψ_g and b_g generated as specified in model (4). The shape and scale parameter values in the Inverse-Gamma distribution were chosen such that $\sigma_g^2, g = 1, .., G$, were generated with mean 10 and moderate variation among the G genes. We set p, the non-null proportion, and ν, the variance factor such

that $p \in \{0.05, 0.25\}, \nu \in \{\frac{1}{2}, 2\}$. Simulations with various sets of parameter values were investigated, but these values lead to representative results among our experiments. For classification, a $N(0, 100)$ prior for τ, $IG(0.1, 0.1)$ prior for ν and for $\sigma^2_{\epsilon,g}$, and a $Beta(1, 1)$ prior for the non-null proportion p were used. In the VB-LIMMA algorithm, the error tolerance was set to be 10^{-6}. The starting values were $\widehat{A_\nu} = \widehat{B_\nu} = 0.1, \sigma^2_{\epsilon,g} \equiv 1, \widehat{M_{\psi_g}} \equiv 0$, and $\widehat{M_{b_g}} = 1$ for genes that are associated with the 5% largest values of d_g or the 5% smallest values of d_g, and $\widehat{M_{b_g}} = 0$ otherwise. For each gene g, $\widehat{M_{b_g}}$ was the VB-approximated posterior mean of the latent indicator b_g, i.e. the gene-specific posterior non-null probability. Hence, for detection of non-null genes, gene ranking based on $\widehat{M_{b_g}}$ can be compared with gene ranking based on the B-statistic, or equivalently gene-specific posterior non-null probability computable from the B-statistic, produced by `limma`.

We note that, varying the value of ν while keeping $\sigma^2_{\epsilon,g}$ unchanged in simulation varies the level of difficulty of the classification problem based on the simulated data. This is because, the B-LIMMA model specifies that the null component $N(\tau, \sigma^2_{\epsilon,g} c_g)$ differs from the non-null component $N(\tau, \sigma^2_{\epsilon,g} c_g + \nu \sigma^2_{\epsilon,g})$ only in variance, and this difference is governed by the multiplicative factor ν. Therefore, data simulated with large ν tends to include non-null genes that are far away from mean τ and thus clearly distinguishable from null genes. The less overlap of the non-null and null component distributions the easier the task of sparse classification for either method. This is reflected in improved ROC curves for both methods as ν is changed from 0.5 to 2 in Figures 2 and 3. In fact, in Figures 2 and 3 we also see that for simulated data with $p \in \{0.05, 0.25\}, \nu \in \{\frac{1}{2}, 2\}$, classification by VB-LIMMA is almost identical to classification by `limma` in terms of ROC curve performance. A cutoff value is defined such that if $\widehat{M_{b_g}} \geq$ cutoff then gene g is classified as non-null. Accuracy is defined as $\frac{TP+TN}{P+N}$ and false discovery rate (FDR) as $\frac{FP}{TP+FP}$, as in Sing et al. (2007). Focusing on a practical range of cutoff values, i.e. $(0.5, 1)$, in Figures 2 and 3, we see that accuracy and FDR of VB are both close to those of `limma`, and that VB has higher accuracy than `limma`. In summary, our experiment suggests that VB-LIMMA, the fully Bayesian classifier, acts as

a comparable classifier to `limma`.

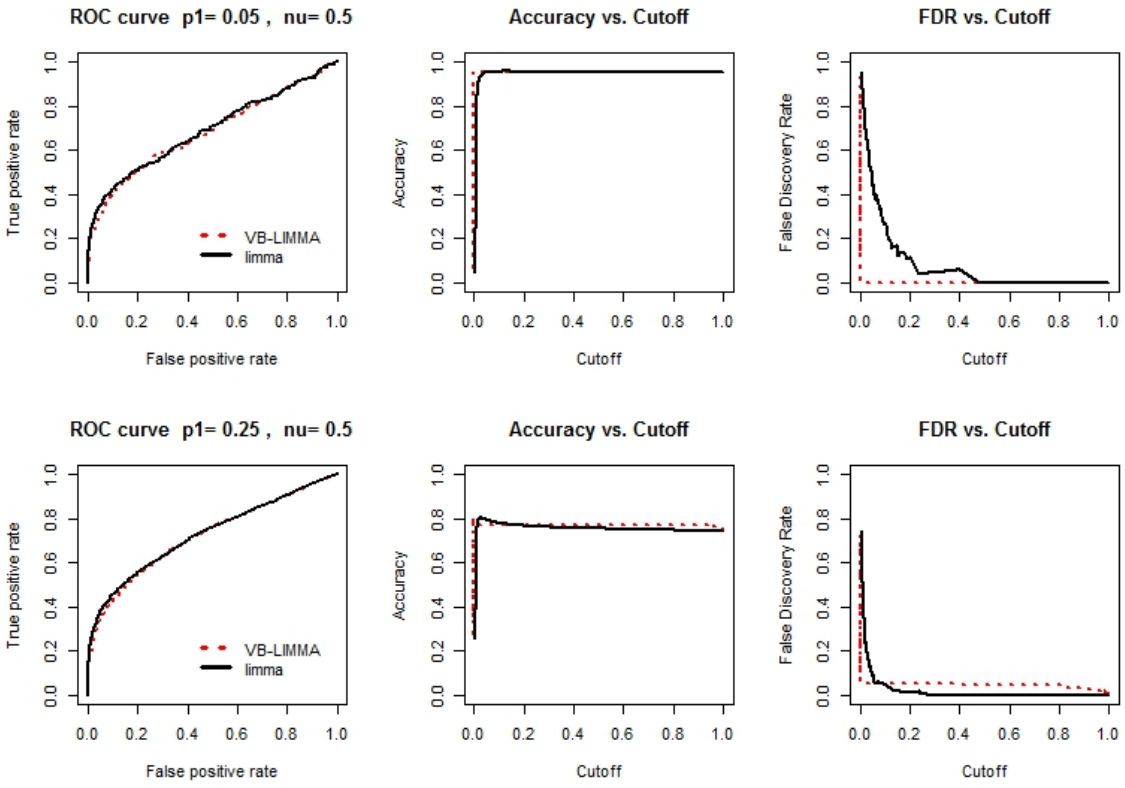

Figure 2: `limma` and VB-LIMMA classification comparison on simulated two-sample data with $G = 5000, n_{1g} \equiv n_{2g} \equiv 8, \tau = 0, p \in \{0.05, 0.25\}, \nu = \frac{1}{2}, \sigma_g^2 \sim IG(41, 400)$ $(i.i.d.)$.

4.1.2 A mixture model extended from the LIMMA model

The two-component mixture model in Bar et al. (2010) differs from the LIMMA model in the assumption on the gene-specific treatment effect ψ_g, specifically

$$\psi_g | \psi, \sigma_\psi^2 \sim N(\psi, \sigma_\psi^2) \quad (i.i.d.) \tag{5}$$

where for the random effect the mean ψ is allowed to be non-zero and the variance σ_ψ^2 is assumed to be independent of the gene-specific error variance. This model is referred to as the LEMMA model in Bar et al. (2010). The LEMMA and LIMMA models both facilitate simultaneous testing of treatment effects on a large number of genes by "borrowing strength" across the genes. In Bar et al. (2010), an empirical Bayes approach is adopted in which parameter estimation of the prior on the error variance $\sigma_{\epsilon,g}^2$ is accomplished through

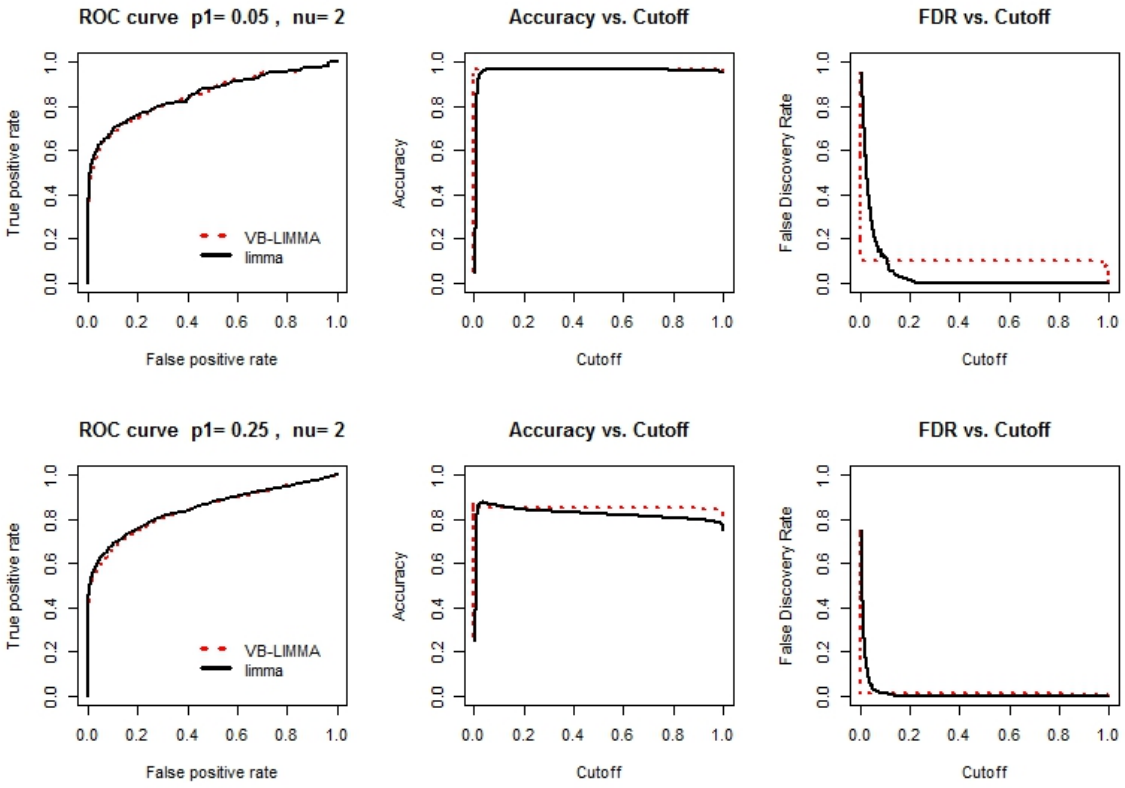

Figure 3: `limma` and VB-LIMMA classification comparison on simulated two-sample data with $G = 5000, n_{1g} \equiv n_{2g} \equiv 8, \tau = 0, p \in \{0.05, 0.25\}, \nu = 2, \sigma_g^2 \sim IG(41, 400)$ $(i.i.d.)$.

maximum marginal likelihood and point estimates of the global parameters are obtained via the EM algorithm. In the EM algorithm, evaluation of the complete data likelihood involves integrating over the prior on the error variance $\sigma^2_{\epsilon,g}$ which is achieved via a Laplace approximation. In what follows we consider a natural extension to the LEMMA model: a fully Bayesian three-component mixture model, B-LEMMA. Formulation of the B-LEMMA model and details of the corresponding VB-LEMMA algorithm are similar to model (4) and what is outlined in Section 3.2, and are included in supplementary materials.

The computational method in Bar et al. (2010) for estimation via Laplace approximation and EM algorithm is implemented in the `lemma` R package (Bar and Schifano, 2010). We consider an MCMC procedure for the B-LEMMA model so that the performance of VB-LEMMA for classification is assessed through comparison with MCMC-LEMMA with reference to `lemma`.

A data set containing $G = 5000$ genes was simulated under the B-LEMMA model using $n_{1g} \equiv 6, n_{2g} \equiv 8, \tau = 0, p_1 = p_2 = 0.1, \psi = 20, \sigma^2_\psi = 2, \sigma^2_g \sim IG(41, 400)$ *i.i.d.*, and ψ_g was generated based on (5). In the VB-LEMMA algorithm, the error tolerance was set to be 10^{-6}. Starting values were set under the same scheme as the VB-LIMMA algorithm, except that the parameter ν was replaced by ψ. The posterior mean of ψ, $\widehat{M_\psi}$, was initialized as the average of the absolute difference between the mean of d_g and the mean of the 5% largest d_g, and the absolute difference between the mean of d_g and the mean of the 5% smallest d_g. Non-informative independent priors were assigned to the global parameters, so that τ and ψ each had a $N(0, 100)$ prior, σ^2_ψ and σ^2_g each had an $IG(0.1, 0.1)$ prior, and $(p_1, p_2, 1-p_1-p_2)$ had a $Dirichlet(1, 1, 1)$ prior. For comparison of classification performance, we used the same prior distributions and starting values in MCMC-LEMMA as in VB-LEMMA. MCMC-LEMMA was run with 1 chain of length 1,200,000, a burn-in period 1,195,000, and a thinning factor 10. We made this choice of chain length to ensure chain convergence after experimenting with various chain settings.

Because point estimates suffice for classification, we present VB-LEMMA and MCMC-LEMMA classification results using posterior means computed from the approximate posterior marginal densities. Genes are classified with a prescribed cutoff value. That is, based on the mean from the approximate marginal posterior of (b_{1g}, b_{2g}) outputted by the VB-LEMMA algorithm, $\widehat{M_{b_{1g}}} \geq$ cutoff $\iff$ gene g belongs to non-null 1 component, and $\widehat{M_{b_{2g}}} \geq$ cutoff $\iff$ gene g belongs to non-null 2 component. The posterior mean of the (b_{1g}, b_{2g}) sample for any gene g is calculable from the MCMC samples and is used to classify gene g with the same cutoff. Point estimates outputted by `lemma` run with default starting values are used as a reference for accuracy of point estimates.

On an Intel Core i5-2430M 2.40GHz, 6GB RAM computer, it took `lemma`, VB-LEMMA, and MCMC-LEMMA 20.94 seconds, 29.24 seconds, and 25353.88 seconds (7 hours) to run, respectively. Table 1 shows the `lemma` point estimates and the approximate marginal posterior means from VB-LEMMA and from MCMC-LEMMA. The parameter estimates from

the three methods are all close to the true values except for σ_ψ^2. The disagreement between the MCMC-LEMMA and VB-LEMMA estimates of σ_ψ^2 is also observable in Figure 4 where the marginal posterior density plots are displayed. In Figure 4, the posterior means of VB-LEMMA and MCMC-LEMMA agree well with each other and with `lemma`, since the vertical lines almost coincide for all parameters except for σ_ψ^2. Deviation of the MCMC-LEMMA estimate of σ_ψ^2 6.993 from the true value 2 renders chain convergence still questionable, despite the long chain we use. Yet we see in Figure 5, which contains the mixture and component densities plots, that VB-LEMMA and MCMC-LEMMA appear to identify the correct non-null 1 and non-null 2 genes since the inward ticks and outward ticks on the x-axis locate at where the two non-null components truly are in the second and fourth plots. True positive rate, TPR, is defined as $\frac{TP1+TP2}{P1+P2}$ and accuracy defined as $\frac{TP1+TP2+TN}{P1+P2+N}$, where $TP1$, $TP2$, TN, $P1$, $P2$, and N are, respectively, number of correctly labeled non-null 1 genes, number of correctly labeled non-null 2 genes, number of correctly labeled null genes, number of true non-null 1 genes, number of true non-null 2 genes, and number of true null genes, such that in this example $P1 + P2 + N = G$. Figure 5 and Table 1 confirm that VB-LEMMA and MCMC-LEMMA classify the genes correctly with high true positive rate and accuracy on the simulated data.

	true	`lemma`	VB-LEMMA	MCMC-LEMMA
τ	0	-0.00650	0.00432	-0.000682
ψ	20	19.764	19.850	19.737
σ_ψ^2	2	5.247	4.278	6.993
p_1	0.1	0.105	0.103	0.105
p_2	0.1	0.105	0.104	0.105
Number of genes in non-null 1	525	520	508	510
Number of genes in non-null 2	524	516	518	518
TPR	NA	0.987	0.977	0.978
Accuracy	NA	0.997	0.995	0.995

Table 1: Posterior estimates, deduced number of non-null genes, and TPR and accuracy for `lemma`, VB-LEMMA, and MCMC-LEMMA on simulated B-LEMMA data with $G = 5000, n_{1g} \equiv 6, n_{2g} \equiv 8, \sigma_g^2 \sim IG(41, 400)$ $(i.i.d.)$ and cutoff= 0.8.

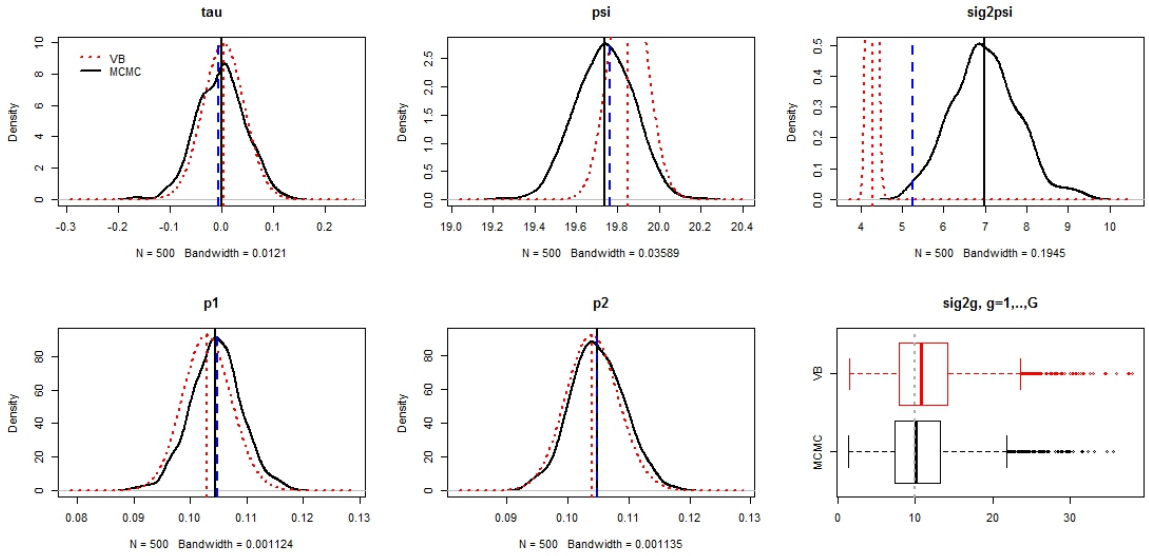

Figure 4: Marginal posterior plots approximated by VB-LEMMA and by MCMC-LEMMA, for the parameters in the B-LEMMA model. For each parameter, the dotted and solid vertical lines represent the estimated posterior mean by VB and by MCMC respectively, and the dashed vertical line marks the estimated parameter value given by `lemma`. The bottom right plot shows the posterior mean of the gene-specific error variance σ_g^2 for each gene g, estimated by VB-LEMMA and by MCMC-LEMMA, and how this quantity varies across the G genes. The dotted line indicates the prior mean 10 of σ_g^2 since the prior distribution of σ_g^2 are i.i.d. $IG(41, 400)$.

The reason why we implement MCMC-LEMMA with only one chain is that we wish to avoid the label-switching issue noted in Section 3.1. For MCMC-LEMMA with one chain, should label-switching occur, ψ would have the opposite sign and the non-null proportions p_1 and p_2 would exchange positions, leading to classification results in which the labels of non-null components 1 and 2 are switched. Thus, with a single chain, the label switching issue is easily resolved. If multiple chains of ψ are run, some will have switched labels. The average of the MCMC simulated samples would then be misleading. Although using one chain requires a much longer chain than when multiple chains are used for reaching convergence, it allows us to distinguish label-switching from non-convergence and easily correct for label-switching in MCMC-LEMMA output.

One approach to address label-switching issue in mixture models is post-processing of MCMC solutions. In the B-LEMMA classification problem, once label-switching is detected,

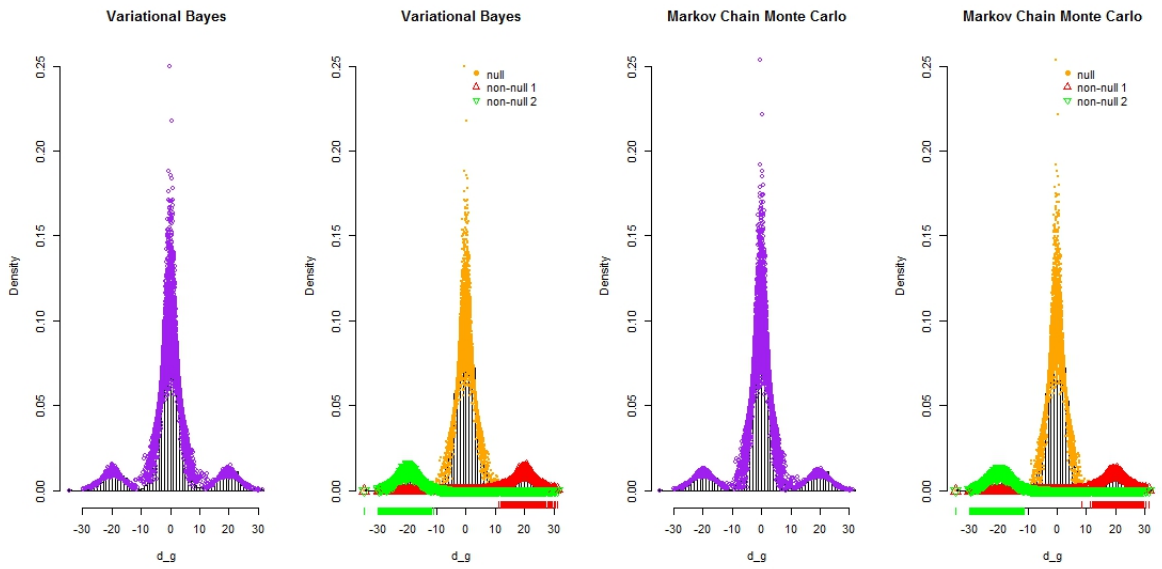

Figure 5: Approximated densities on simulated data with $G = 5000$. From left to right: mixture density approximated by VB-LEMMA, the three component densities approximated by VB-LEMMA, mixture density approximated by MCMC-LEMMA, and the three component densities approximated by MCMC-LEMMA. On the x-axis in the second and the fourth plots, inward ticks indicate the classified non-null 1 genes and outward ticks indicate the classified non-null 2 genes by the corresponding method.

we can post-process MCMC-LEMMA classification results by giving the opposite sign to the MCMC sample of ψ, switching the p_1 sample for the p_2 sample and vice versa, and exchanging the non-null 1 and non-null 2 statuses. To make Figure 6, where performance of VB-LEMMA and MCMC-LEMMA was recorded for 30 simulated data sets, the data simulation settings and the settings for the two methods were kept unchanged from the previous experiment, and MCMC-LEMMA results were post-processed according to the aforementioned scheme wherever label-switching occurred. Posterior means of σ^2_ψ estimated by MCMC-LEMMA persisted as worse over-estimates than those estimated by VB-LEMMA. Nonetheless, the component densities plots for the 30 simulated data sets (not shown) with gene labels all appear similar to those in Figure 5, indicating that both VB-LEMMA and MCMC-LEMMA are reliable classifiers in this context. However, VB-LEMMA is far superior in terms of computational efficiency.

Another way of avoiding label-switching issue in mixture models is assigning a prior

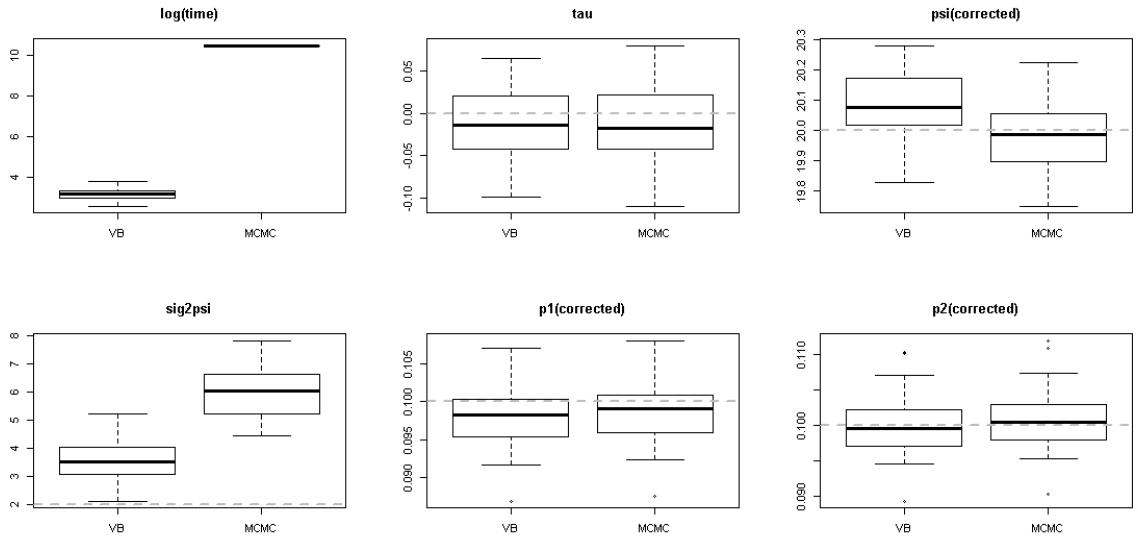

Figure 6: Comparison between VB-LEMMA and MCMC-LEMMA with multiple simulation on an Intel Xeon L5410 2.33GHz, 8GB RAM computer. 30 data sets were simulated with same true parameter values in Table 1. MCMC-LEMMA results were post-processed so that results were corrected for label-switching among the simulated data sets. The dashed lines mark the true parameter values.

distribution on the parameter that appropriately restricts the sampling process in MCMC, as Christensen et al. (2011) points out. Although the mixture and component densities plots (not shown) show that MCMC-LEMMA, when implemented with a half-normal prior on ψ, classifies the genes satisfactorily, restricting the mean of ψ to a positive value changes the marginal data distribution of the B-LEMMA model. Therefore, direct comparisons with the modified MCMC-LEMMA are no longer valid because VB-LEMMA and MCMC-LEMMA with a half-normal prior on ψ are based on two different models.

These B-LEMMA simulation experiments indicate that VB-LEMMA efficiently and accurately approximates fully Bayesian estimation and classification results. In contrast, MCMC-LEMMA, although theoretically capable of achieving extremely high accuracy, requires substantially more work in overcoming issues such as non-convergence and label-switching.

4.1.3 A mixture model for count data

In mass-spectrometry-based shotgun proteomics, spectral counts measure abundance of proteins under various conditions. By comparing spectral counts under control and treatment conditions, researchers can test for differential abundance simultaneously for a large number of proteins. Booth et al. (2011) propose a mixture of log-linear models with random protein specific factors for classifying proteins into null and non-null (differential abundance) categories. Their model, which is fully Bayesian, can be written as follows:

$$y_{ij}|\mu_{ij} \sim Poisson(\mu_{ij}), \tag{6}$$

$$\log \mu_{ij}|I_i, \beta_0, \beta_1, b_{0i}, b_{1i} = \beta_0 + b_{0i} + \beta_1 T_j + b_{1i} I_i T_j + \beta_2 I_i T_j + \log L_i + \log N_j,$$

$$I_i|\pi_1 \sim Bernoulli(\pi_1) \ \ i.i.d., \qquad b_{ki}|\sigma_k^2 \sim N(0, \sigma_k^2) \ \ i.i.d., \ \ k = 0, 1,$$

$$\beta_m \sim N(0, \sigma_{\beta_m}^2), \ \ m = 0, 1, 2, \qquad \sigma_k^{-2} \sim Gamma(A_{\sigma_k^2}, B_{\sigma_k^2}), \ \ k = 0, 1, \qquad \pi_1 \sim Beta(\alpha, \beta),$$

where y_{ij} is the spectral count of protein i, $i = 1, .., p$, and replicate j, $j = 1, .., n$. L_i is the length of protein i, N_j is the average count for replicate j over all proteins, and

$$T_j = \begin{cases} 1 & \text{if replicate } j \text{ is in the treatment group} \\ 0 & \text{if replicate } j \text{ is in the control group.} \end{cases}$$

In fact, conjugacy in this Poisson GLMM is not sufficient for a tractable solution to be computed by VB. Therefore, a similar Poisson-Gamma HGLM where the parameters $\beta_m, m = 0, 1, 2$ and the latent variables $b_{ki}, k = 0, 1$ are transformed is used for the VB implementation:

$$y_{ij}|\mu_{ij} \sim Poisson(\mu_{ij}), \tag{7}$$

$$\log \mu_{ij}|I_i, \beta_0, \beta_1, b_{0i}, b_{1i} = \beta_0 + b_{0i} + \beta_1 T_j + b_{1i} I_i T_j + \beta_2 I_i T_j + \log L_i + \log N_j,$$

$$I_i|\pi_1 \sim Bernoulli(\pi_1) \ \ i.i.d., \qquad b_{ki}|\phi_{ki} = \log(\phi_{ki}^{-1}), \ \ k = 0, 1,$$

$$\phi_{ki}|\delta_k \sim IG(\delta_k, \delta_k) \ \ i.i.d., \ k = 0, 1, \qquad \delta_k \sim Gamma(A_{\delta_k}, B_{\delta_k}), \ \ k = 0, 1,$$

$$\beta_m|\lambda_m = \log(\lambda_m^{-1}), \ \ m = 0, 1, 2, \qquad \lambda_m \sim IG(A_{\lambda_m}, B_{\lambda_m}), \ \ m = 0, 1, 2, \qquad \pi_1 \sim Beta(\alpha, \beta).$$

As before classification is inferred from the posterior expectations of the latent binary indicators $I_i, i = 1, ..., p$. This fully Bayesian, two-component mixture model allows for

derivation of a VB algorithm, VB-proteomics, the details of which are shown in supplementary materials.

For any real data, the exact model that the data distribution follows is unknown. To imitate challenges in sparse classification on real data sets, we simulated data from each of the two models: Poisson GLMM (6) and Poisson-Gamma HGLM (7), and applied VB-proteomics to both data sets. MCMC-proteomics, also derived based on Poisson-Gamma HGLM (7), was implemented via OpenBUGS in R with a chain of length 600,000, a burn-in period 595,000, and a thinning factor 5. Starting values implemented in VB-proteomics were consistent with those in MCMC-proteomics.

Two data sets, one following the Poisson GLMM (6) with $\beta_0 = -7.7009$, $\beta_1 = -0.1765$, $\beta_2 = 0$, $\sigma_0^2 = 1$, $\sigma_1^2 = 4$ and the other following the Poisson-Gamma HGLM (7) with $\lambda_0 = 2210.336$, $\lambda_1 = 1.193$, $\lambda_2 = 1$, $\delta_0 = 1$, $\delta_1 = 0.2$ were generated. In each data set there were 4 replicates under each treatment and $p = 1307$ proteins. The non-null indicator was generated with $\pi_1 = 0.2$. These values were chosen so that the simulated data sets were similar to the Synthetic 2-fold data analyzed in Booth et al. (2011).

	Poisson GLMM			Poisson-Gamma HGLM		
	true	VB	MCMC	true	VB	MCMC
number of non-nulls	259	109	96	262	103	90
π_1	0.2	0.0856	0.160	0.2	0.122	0.218
accuracy		0.878	0.874		0.871	0.865
time		54 seconds	7.5 hours		10 seconds	7.4 hours

Table 2: Proteomics count data: simulation under the two models and VB-proteomics and MCMC-proteomics classifications. Total number of proteins is 1307. Cutoff is 0.8. Accuracy is defined in Section 4.1.1.

Table 2 gives the classification and computational performance of VB-proteomics and MCMC-proteomics for the two simulated data sets on an Intel Core i5-2430M 2.40GHz, 6GB RAM computer. The longer running time and the larger deviation of posterior mean of π_1 from its true value associated with the simulated data under Poisson GLMM in Table 2 shows that it is indeed more difficult for VB-proteomics to perform classification on data

from the "wrong" model. Nonetheless, VB-proteomics identifies the extreme proteins for both data sets with similar accuracy regardless of which model the data was generated from, and is much more desirable than MCMC-proteomics in terms of computational speed.

Figures 7 and 8 show plots of the log ratios of total counts (+1) in treatment and control groups against protein number, and ROC curves comparing performance of the model-based VB-proteomics, MCMC-proteomics approaches with one protein at a time Score tests. These plots clearly indicate that VB-proteomics acts as a better classifier than individual Score tests and as a comparable classifier to MCMC-proteomics in terms of ROC curve, accuracy, and false discovery rate performance.

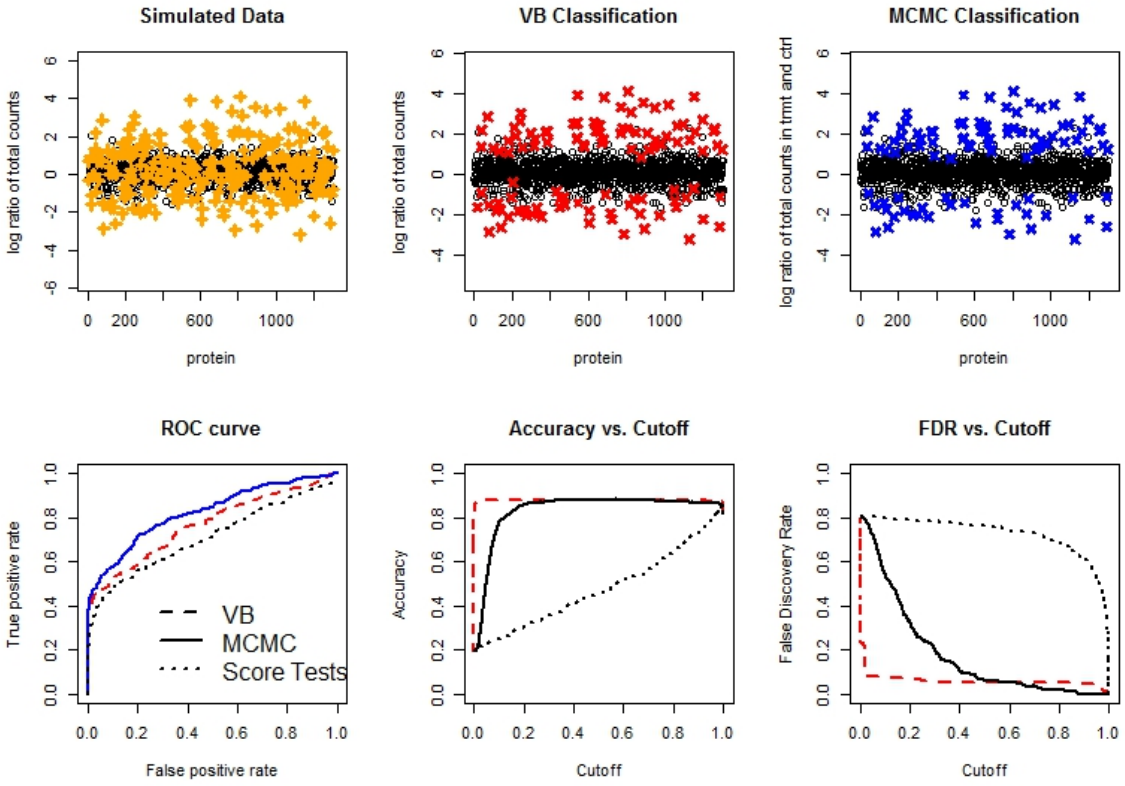

Figure 7: Simulated data under the Poisson GLMM (6), i.e. the "wrong" model. Top row: plots of log ratios of total counts in treatment and control groups against protein number with true non-null proteins, VB-proteomics labeled proteins, and MCMC-proteomics labeled proteins indicated. Bottom row: ROC curves, accuracy vs. cutoff, and FDR vs. cutoff for VB-proteomics, MCMC-proteomics, and for one protein at a time score tests. Accuracy and FDR are as defined in Section 4.1.1.

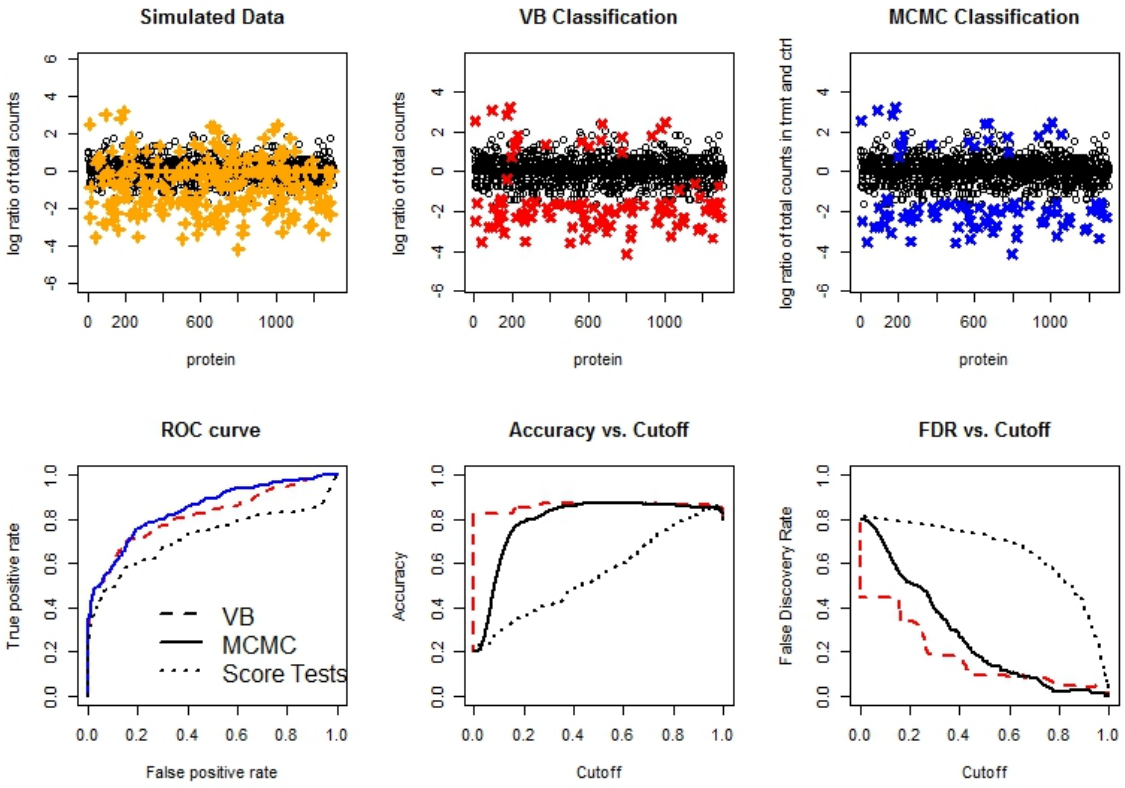

Figure 8: Simulated data under the Poisson-Gamma HGLM (7). Top row: plots of log ratios of total counts in treatment and control groups against protein number with true non-null proteins, VB-proteomics labeled proteins, and MCMC-proteomics labeled proteins indicated. Bottom row: ROC curves, accuracy vs. cutoff, and FDR vs. cutoff for VB-proteomics, MCMC-proteomics, and for one protein at a time score tests. Accuracy and FDR are as defined in Section 4.1.1.

4.2 Real data examples

In this section, we further illustrate classification performance of VB for fully Bayesian finite mixture models on real data. Two microarray examples are investigated, the APOA1 (Callow et al., 2000) and Colon Cancer (Alon et al., 1999) data sets.

4.2.1 APOA1 data

The APOA1 data (Callow et al., 2000) contains 5548 genes associated with measurements from 8 control mice and 8 "knockout" mice. The data was originally obtained from a two-groups design. The histogram of d_g in Figure 9 indicates that the B-LIMMA model (4) is

a good fit and that both `limma` and VB-LIMMA are applicable for classification, under the assumption that majority of the genes are null. The small proportion of the non-null genes is known since there are eight true non-null genes. Figure 10 shows the top 10 non-null genes by `limma`. VB-LIMMA was implemented, with non-informative priors and starting values such that $\widehat{M_{b_g}} = 1$ for genes that are associated with the 5% largest values of d_g or the 5% smallest values of d_g and $\widehat{M_{b_g}} = 0$ otherwise. VB-LIMMA identified the same top eight genes as `limma` in 53 seconds, indicated by close-to-one posterior non-null probability estimate of those genes in Figure 9. The `convest` function in `limma` gave an over-estimate of the non-null proportion p, 0.130114, whereas VB-LIMMA produced a closer estimate 0.00162162 to the true value 0.00144.

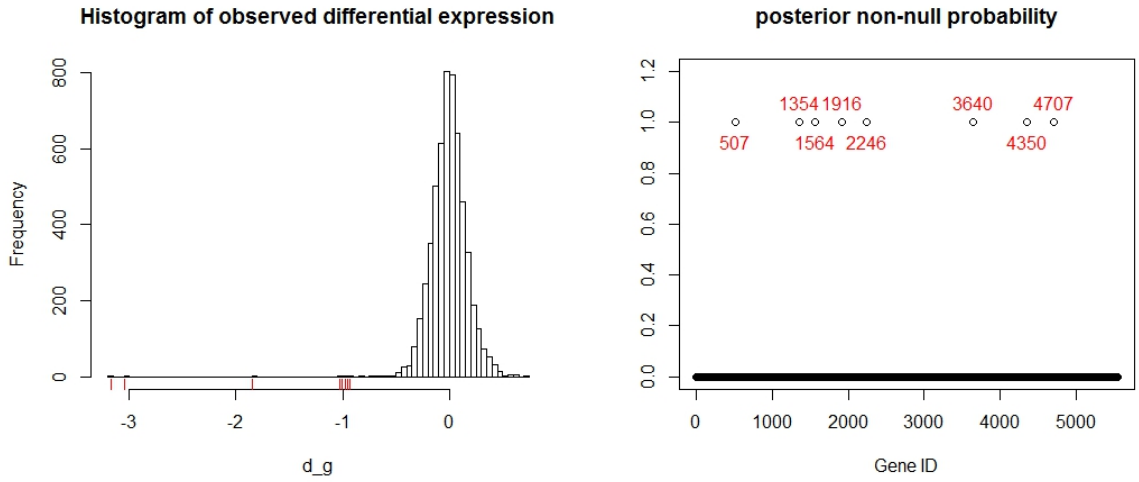

Figure 9: APOA1 data: histogram of d_g with the eight non-null genes classified by VB-LIMMA indicated by inward ticks on the x-axis, and posterior mean of b_g (i.e. posterior non-null probability of gene g) estimated by VB-LIMMA with the eight non-null gene numbers identified.

4.2.2 Colon Cancer data

The Colon Cancer data (Alon et al., 1999) consists of gene expression values from 2000 genes in 22 controls and 40 treatment samples. Since the two-component mixture model is a special case of the three-component mixture model with one of the non-null proportions

	NAME	logFC	t	P.Value	adj.P.Val	B
1916	ApoAI,lipid-Img	-3.1661645	-24.048796	5.442072e-15	3.019261e-11	15.960594
507	EST,HighlysimilartoA	-3.0485504	-12.938819	1.797129e-10	4.985235e-07	11.347701
4707	CATECHOLO-METHYLTRAN	-1.8481659	-12.458569	3.300689e-10	6.104074e-07	10.982250
3640	EST,WeaklysimilartoC	-1.0269537	-11.894827	6.905520e-10	9.577957e-07	10.524492
1564	ApoCIII,lipid-Img	-0.9325824	-9.927515	1.145390e-08	1.270925e-05	8.647893
2246	ESTs,Highlysimilarto	-1.0098117	-9.065277	4.461600e-08	3.579128e-05	7.667271
1354	est	-0.9774236	-9.057852	4.515842e-08	3.579128e-05	7.658359
4350	similartoyeaststerol	-0.9549693	-7.465666	7.068872e-07	4.902263e-04	5.546823

Figure 10: `limma` output: Top 10 genes detected as non-null for APOA1 data. Gene numbers are shown as row names.

being zero, both B-LIMMA and B-LEMMA models are suitable for classification of genes on the Colon Cancer data. Therefore, `limma`, VB-LIMMA, `lemma`, and VB-LEMMA were implemented.

A gene rankings table showing genes that are associated with the top 200 largest posterior non-null probabilities was produced according to results from each of the four procedures. For all procedures, the posterior non-null probability for any gene equals one minus the posterior null probability of that gene. Comparing the top genes tables reveals that out of the top 200 identified non-null genes, `limma` agrees with VB-LIMMA on 151 genes, `lemma` agrees with VB-LEMMA on 190 genes, and all four methods share 140 genes. This high level of agreement between the procedures shows that fully Bayesian classification via VB is comparable in accuracy to classification via empirical Bayes approaches on the Colon Cancer data.

MCMC-LEMMA was also implemented for the Colon Cancer data to illustrate performance of MCMC for classification on real data under the B-LEMMA model. A single chain of length 4,000,000, a burn-in period 3,995,000, and a thinning factor 10 was used to ensure convergence and avoid label-switching. MCMC diagnostics plots (not shown) suggests that convergence is tolerable, although it took MCMC-LEMMA 11.7 hours to run on an Intel Xeon L5410 2.33GHz, 8GB RAM computer. There are 187 genes present in both top 200 genes tables based on MCMC-LEMMA and VB-LEMMA classifications, which shows accuracy of VB in classification is comparable with that of MCMC. From the component densities plots in Figure 11 we see that, the three-component mixture evaluated by VB-LEMMA and

by MCMC-LEMMA are two different yet plausible solutions based on the B-LEMMA model. It is unknown which solution is closer to the truth. Nonetheless, with a 0.9 cutoff, MCMC-LEMMA did not detect any non-null genes, whereas VB-LEMMA identified 56 group 1 non-null genes and 30 group 2 non-null genes in 7 seconds.

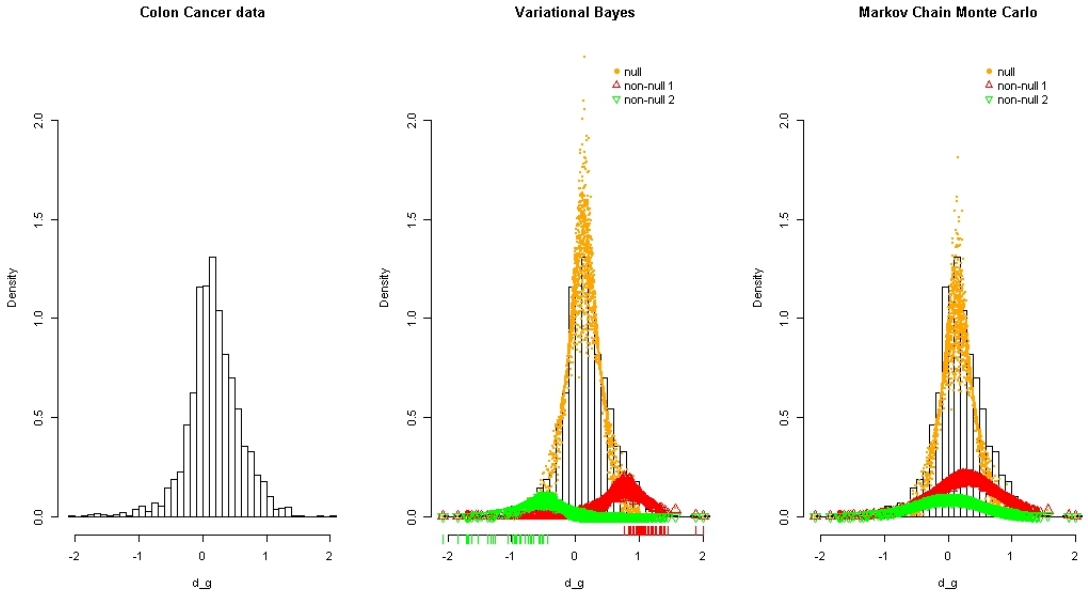

Figure 11: From left to right: Histogram of d_g for Colon Cancer data, component densities estimated by VB-LEMMA, and component densities estimated by MCMC-LEMMA. On the x-axis, inward ticks indicate the classified non-null 1 genes and outward ticks indicate the classified non-null 2 genes with a 0.9 cutoff.

5 Discussion

VB has been promoted in the statistical and computer science literature as an alternative to MCMC for Bayesian computation. However, the performance and feasibility of VB has not been widely investigated for hierarchical mixture models used in sparse classification problems. This article demonstrates the implementation of VB in that context and shows that it is capable of efficiently producing reliable solutions with classification performance comparable to much slower MCMC methods. A known issue with VB is its tendency to underestimate variability in marginal posteriors. GBVA is designed to correct this, but its

performance in the context of hierarchical mixture models is not promising either in terms of accuracy or computational speed.

As the demand for high-dimensional data analysis tools grow the search for fast and accurate alternatives to MCMC continues to be an important open research area. Other deterministic alternatives to MCMC in posterior density approximation include EP (Minka, 2001b,a) and INLA (Rue et al., 2009). In EP, while similar factorization of the joint posterior density to that in VB is assumed, the approximate posterior is found by moment matching implied by minimization of the reverse form of Kullback-Leibler divergence to that used in VB theory. However, as noted in Bishop (2006), drawbacks of EP include lack of guarantee of convergence and failure to capture any mode if the true posterior density is multimodal. For the class of latent Gaussian models, approximate Bayesian inference can be achieved via INLA (Rue et al., 2009). The INLA method relies on Gaussian approximations and numerical methods and bypasses any assumption of factorized density forms. A limitation of INLA is the fact that models that contain non-Gaussian latent variables in the linear predictor do not belong to the class of latent Gaussian models for which INLA is applicable.

VB has been criticized (Rue et al., 2009) for its applicability being restricted to the class of conjugate-exponential (Beal, 2003) models. However, provided that VB is feasible for a reparameterized model whose inference is close to the original model (Beal, 2003), VB is often a good approach for fully Bayesian inference because of its relative ease of implementation and computational speed.

Acknowledgments

We would like to thank John T. Ormerod who provided supplementary materials for GBVA implementation in Ormerod (2011), and Haim Y. Bar for helpful discussions.

Supplementary Materials

VB-LEMMA and VB-proteomics algorithms: Details of the VB-LEMMA and VB-proteomics algorithms, i.e. VB implementation for the B-LEMMA model in Section 4.1.2 and proteomics Poisson-Gamma HGLM in Section 4.1.3. (PDF file)

References

Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., and Levine, A. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750.

Attias, H. (2000). A variational Bayesian framework for graphical models. *Advances in Neural Information Processing Systems*, 12(1-2):209–215.

Bar, H., Booth, J., Schifano, E., and Wells, M. (2010). Laplace approximated EM microarray analysis: an empirical Bayes approach for comparative microarray experiments. *Statistical Science*, 25(3):388–407.

Bar, H. and Schifano, E. (2010). Lemma: Laplace approximated EM microarray analysis. R package version 1.3-1. `http://CRAN.R-project.org/package=lemma`.

Beal, M. (2003). *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, University of London.

Bishop, C. (1999). Variational principal components. In *Proceedings of Ninth International Conference on Artificial Neural Networks, ICANN'99*, volume 1, pages 509–514. IET.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer Science+ Business Media, New York.

Bishop, C., Spiegelhalter, D., and Winn, J. (2002). VIBES: A variational inference engine for Bayesian networks. *Advances in Neural Information Processing Systems*, 15:777–784.

Blei, D. and Jordan, M. (2006). Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143.

Booth, J., Eilertson, K., Olinares, P., and Yu, H. (2011). A Bayesian mixture model for comparative spectral count data in shotgun proteomics. *Molecular & Cellular Proteomics*, 10(8).

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Callow, M., Dudoit, S., Gong, E., Speed, T., and Rubin, E. (2000). Microarray expression profiling identifies genes with altered expression in HDL-deficient mice. *Genome Research*, 10(12):2022–2029.

Christensen, R., Johnson, W. O., Branscum, A. J., and Hanson, T. E. (2011). *Bayesian Ideas and Data Analysis: An Introduction for Scientists and Statisticians*. CRC Press, Boca Raton.

Consonni, G. and Marin, J. (2007). Mean-field variational approximate Bayesian inference for latent variable models. *Computational Statistics & Data Analysis*, 52(2):790–798.

Corduneanu, A. and Bishop, C. (2001). Variational Bayesian model selection for mixture distributions. In *Artificial Intelligence and Statistics 2001*, Tommi S. Jaakkola and Thomas S. Richardson (Eds.), pages 27–34. Morgan Kaufmann, Waltham.

Cowles, M. K. and Carlin, B. P. (1996). Markov chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, 91(434):883–904.

De Freitas, N., Højen-Sørensen, P., Jordan, M., and Russell, S. (2001). Variational MCMC. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, John Breese and Daphne Koller (Eds.), pages 120–127. Morgan Kaufmann Publishers Inc., San Francisco.

Efron, B. (2008). Microarrays, empirical Bayes and the two-groups model. *Statistical Science*, 23(1):1–22.

Faes, C., Ormerod, J., and Wand, M. (2011). Variational Bayesian inference for parametric and nonparametric regression with missing data. *Journal of the American Statistical Association*, 106(495):959–971.

Friston, K., Ashburner, J., Kiebel, S., Nichols, T., and Penny, W. (2011). *Statistical Parametric Mapping: The Analysis of Functional Brain Images*. Academic Press, London.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2003). *Bayesian Data Analysis*. Chapman & Hall/CRC.

Ghahramani, Z. and Beal, M. (2000). Variational inference for Bayesian mixtures of factor analysers. *Advances in Neural Information Processing Systems*, 12:449–455.

Goldsmith, J., Wand, M., and Crainiceanu, C. (2011). Functional regression via variational Bayes. *Electronic Journal of Statistics*, 5:572.

Grimmer, J. (2011). An introduction to Bayesian inference via variational approximations. *Political Analysis*, 19(1):32–47.

Honkela, A. and Valpola, H. (2005). Unsupervised variational Bayesian learning of nonlinear models. In *Advances in Neural Information Processing Systems 17*, Lawrence K. Saul, Yair Weiss and Léon Bottou (Eds.), pages 593–600. MIT Press.

Jaakkola, T. S. (2000). Tutorial on variational approximation methods. In *Advanced Mean Field Methods: Theory and Practice*, Manfred Opper and David Saad (Eds.), pages 129–159. MIT Press.

Li, J., Das, K., Fu, G., Li, R., and Wu, R. (2011). The Bayesian lasso for genome-wide association studies. *Bioinformatics*, 27(4):516–523.

Li, Z. and Sillanpää, M. (2012). Estimation of quantitative trait locus effects with epistasis by variational

Bayes algorithms. *Genetics*, 190(1):231–249.

Logsdon, B., Hoffman, G., and Mezey, J. (2010). A variational Bayes algorithm for fast and accurate multiple locus genome-wide association analysis. *BMC Bioinformatics*, 11(1):58.

Luenberger, D. and Ye, Y. (2008). *Linear and Nonlinear Programming*, volume 116 of *International Series in Operations Research & Management Science*. Springer, New York.

Marin, J.-M. and Robert, C. P. (2007). *Bayesian Core: A Practical Approach to Computational Bayesian Statistics*. Springer, New York.

Martino, S. and Rue, H. (2009). R package: INLA. *Department of Mathematical Sciences, NTNU, Norway. Available at http://www.r-inla.org*.

McGrory, C. and Titterington, D. (2007). Variational approximations in Bayesian model selection for finite mixture distributions. *Computational Statistics & Data Analysis*, 51(11):5352–5367.

McLachlan, G. and Peel, D. (2004). *Finite Mixture Models*. Wiley, New York.

Minka, T. (2001a). Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, John Breese and Daphne Koller (Eds.), pages 362–369. Morgan Kaufmann Publishers Inc., San Francisco.

Minka, T. (2001b). *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Massachusetts Institute of Technology.

Ormerod, J. (2011). Grid based variational approximations. *Computational Statistics & Data Analysis*, 55(1):45–56.

Ormerod, J. and Wand, M. (2010). Explaining variational approximations. *The American Statistician*, 64(2):140–153.

Rue, H., Martino, S., and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B*, 71(2):319–392.

Salter-Townshend, M. and Murphy, T. (2009). Variational Bayesian inference for the latent position and cluster model. In *NIPS 2009 (Workshop on Analyzing Networks & Learning with Graphs)*.

Sing, T., Sander, O., Beerenwinkel, N., and Lengauer, T. (2007). ROCR: Visualizing the performance of scoring classifiers. R package version 1.0-2. `http://rocr.bioinf.mpi-sb.mpg.de/ROCR.pdf/`.

Smídl, V. and Quinn, A. (2005). *The Variational Bayes Method in Signal Processing*. Springer-Verlag, Berlin.

Smyth, G. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*. Volume 3, Issue 1, Article 3.

Smyth, G. (2005). Limma: linear models for microarray data. *Bioinformatics and Computational Biology Solutions using R and Bioconductor*, pages 397–420. Springer, New York.

Teschendorff, A., Wang, Y., Barbosa-Morais, N., Brenton, J., and Caldas, C. (2005). A variational Bayesian mixture modelling framework for cluster analysis of gene-expression data. *Bioinformatics*, 21(13):3025–3033.

Tzikas, D., Likas, A., and Galatsanos, N. (2008). The variational approximation for Bayesian inference. *Signal Processing Magazine, IEEE*, 25(6):131–146.

Wand, M. P., Ormerod, J. T., Padoan, S. A., and Frührwirth, R. (2011). Mean field variational Bayes for elaborate distributions. *Bayesian Analysis*, 6(4):1–48.

Wang, B. and Titterington, D. M. (2005). Inadequacy of interval estimates corresponding to variational Bayesian approximations. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, Robert G. Cowell and Zoubin Ghahramani (Eds.), pages 373–380. Society for Artificial Intelligence and Statistics.

Zhang, M., Montooth, K., Wells, M., Clark, A., and Zhang, D. (2005). Mapping multiple quantitative trait loci by Bayesian classification. *Genetics*, 169(4):2305–2318.