Latent-Space Variational Bayes

Jaemo Sung, *Student Member*, *IEEE*,
Zoubin Ghahramani, *Member*, *IEEE*, and
Sung-Yang Bang

Abstract—Variational Bayesian Expectation-Maximization (VBEM), an approximate inference method for probabilistic models based on factorizing over latent variables and model parameters, has been a standard technique for practical Bayesian inference. In this paper, we introduce a more general approximate inference framework for *conjugate-exponential* family models, which we call *Latent-Space Variational Bayes (LSVB)*. In this approach, we integrate out the model parameters in an exact way, leaving only the latent variables. It can be shown that the LSVB approach gives better estimates of the model evidence as well as the distribution over the latent variables than the VBEM approach, but, in practice, the distribution over the latent variables has to be approximated. As a practical implementation, we present a *First-order LSVB (FoLSVB)* algorithm to approximate the distribution over the latent variables. From this approximate distribution, one can also estimate the model evidence and the posterior over the model parameters. The FoLSVB algorithm is directly comparable to the VBEM algorithm and has the same computational complexity. We discuss how LSVB generalizes the recently proposed collapsed variational methods to general conjugate-exponential families. Examples based on mixtures of Gaussians and mixtures of Bernoullis with synthetic and real-world data sets are used to illustrate some advantages of our method over VBEM.

Index Terms—Bayesian inference, conjugate-exponential family, variational method, mixture of Gaussians, mixture of Bernoullis.

1 Introduction

BAYESIAN approaches have drawn attention in machine learning and statistical analysis in recent decades since they have advantages compared with maximum likelihood approaches. Bayesian methods do not suffer from overfitting (since they do not involve fitting parameters) and provide a coherent approach to averaging over as well as comparing models.

Assuming a model $\mathcal{M}$, in the Bayesian approach, all inferences are automatically done by applying Bayes' rule starting from a prior $P(\boldsymbol{\theta}|\mathcal{M})$ over model parameters. For an example, we can obtain a posterior distribution over latent variables X and model parameters $\boldsymbol{\theta}$ given data set Y:

$$P(X,\boldsymbol{\theta}|Y,\mathcal{M}) = \frac{P(Y,X|\boldsymbol{\theta},\mathcal{M})P(\boldsymbol{\theta}|\mathcal{M})}{P(Y|\mathcal{M})}, \tag{1}$$

$$P(Y|\mathcal{M}) = \int dXd\boldsymbol{\theta}P(Y,X|\boldsymbol{\theta},\mathcal{M})P(\boldsymbol{\theta}|\mathcal{M}). \tag{2}$$

The posterior distribution $P(X,\boldsymbol{\theta}|Y,\mathcal{M})$ is useful for cluster analysis, dimensionality reduction, classification, and prediction tasks. In particular, the probability $P(Y|\mathcal{M})$, called the *marginal likelihood* or *model evidence*, has been shown to penalize overcomplex models by automatically encoding Occam's Razor [1], [2], so it is important for model comparison [3]. A more in-depth introduction to Bayesian approaches is given in [4], [5], [6] and the references therein.

Unfortunately, for many interesting latent variable models, true Bayesian inferences are generally intractable due to the high-dimensional integrals associated with them. Therefore, such difficult integrals have to be approximated for practical Bayesian inferences. There are two standard approximate methods, the Monte Carlo method [7], [8], [9], [10] and the variational method [6], [10], [11], [12]. Markov chain Monte Carlo (MCMC) methods provide asymptotic theoretical guarantees but are often impractical due to their computational cost as well as the difficulty of monitoring convergence. On the other hand, the variational method, which is called *Variational Bayes (VB)* for Bayesian inferences, can require much less computation and comes with an easy to evaluate convergence criterion, but does not have the same asymptotic guarantees as MCMC.

Recently, *Variational Bayesian Expectation-Maximization (VBEM)* [13], [14] has become one of the standard VB approximate inference methods and has been successfully applied to many interesting latent variable models [14], [15], [16], [17], [18], [19]. However, in practice, maximizing over hyperparameters can cause VBEM to be fraught with spurious local maxima and to suffer from the overfitting problem.

In this paper, we consider a more general VB approximate inference framework, which we call *Latent-Space VB (LSVB)*. LSVB is constructed under a weaker independence assumption than VBEM. In the VBEM approach, we assume that the latent variables are independent of the model parameters. This assumption is too strong since the fluctuation of the model parameters directly affects the latent variables in general. Therefore, VBEM inherently ignores some important correlations. In the LSVB approach, we only assume that the latent variables over samples are independent of each other, integrating out the model parameters exactly. This is a more reasonable assumption in the real world since the fluctuation of the latent variables on a single sample exhibits weak influences on the other latent variables via summary statistics. Moreover, such influences from a single sample will eventually be negligible for large data sets. Fundamentally, it can be shown that the LSVB approach achieves better estimates of the model evidence as well as the distribution over the latent variables than the VBEM approach. However, the distribution over the latent variables has to be approximated in practice.

A similar idea, integrating out the model parameters, was simultaneously proposed in [20][1] and extended in [21] and called *collapsed variational (CV) approximations*. The work in these papers was restricted to a specific Dirichlet-Multinomial model. We consider, in our original technical report and here in this paper, a more general class of latent variable models called the *conjugate-exponential family* and propose *First-order LSVB (FoLSVB)*, a new practical VB approximate inference algorithm having the same computational complexity as the VBEM algorithm. Integrating out the model parameters like LSVB, the FoLSVB algorithm sometimes has better convergence properties than the VBEM algorithm.

This paper is organized as follows: In Section 2, we introduce the general idea of LSVB, briefly reviewing VBEM. In Section 3, we apply LSVB to the conjugate-exponential model and derive the FoLSVB algorithm. In Sections 4 and 5, we give examples of the mixture of Gaussians (MoG) and the mixture of Bernoullis (MoB)

- *J. Sung and S.-Y. Bang are with the Department of Computer Science and Engineering, Pohang University of Science and Technology, San 31 Hyoja-dong, Nam-gu, Pohang, Kyungbuk, 790-784, South Korea. E-mail: {emtidi, sybang}@postech.ac.kr.*
- *Z. Ghahramani is with the Information Engineering Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK. E-mail: zoubin@eng.cam.ac.uk.*

Manuscript received 7 Aug. 2007; revised 2 Jan. 2008; accepted 20 May 2008; published online 5 June 2008.
Recommended for acceptance by N. Lawrence.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2007-08-0485.
Digital Object Identifier no. 10.1109/TPAMI.2008.157.

[1] Our work was known to those authors in the form of a technical report.

and use synthetic and real-world data sets to demonstrate our method as compared to VBEM. Finally, we conclude in Section 6.

2 Latent-Space Variational Bayes

Assume a data set $Y = \{\boldsymbol{y}_i\}_{i=1}^N$ and a latent variable set $X = \{\boldsymbol{x}_i\}_{i=1}^N$ of N i.i.d. samples, drawn from a joint distribution $P(\boldsymbol{y}_i, \boldsymbol{x}_i|\boldsymbol{\theta})$ parameterized by model parameters $\boldsymbol{\theta}$. A complete data set consists of Y and X. We allow both $\boldsymbol{y}_i$ and $\boldsymbol{x}_i$ to be multidimensional.

In the VB approximate framework, we form a lower bound of log marginal likelihood, $\log P(Y)$, with respect to an approximating distribution for the posterior distribution and then approximate inferences are done by maximizing the lower bound. For an example, VBEM employs a factorized approximating distribution over the latent variables and the model parameters, such as $Q(X, \boldsymbol{\theta}) = Q_X(X)Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$; the approximating distribution Q_X exhibits additional factorizations over samples, that is, $Q_X(X) = \prod_{i=1}^N Q_{\boldsymbol{x}_i}(\boldsymbol{x}_i)$. From this, we can form a lower bound $\mathcal{F}_{Q_X Q_{\boldsymbol{\theta}}}$ by using Jensen's inequality:

$$\begin{aligned}\log P(Y) \geq & \int dX d\boldsymbol{\theta} Q_X(X) Q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \log \frac{P(Y, X|\boldsymbol{\theta})P(\boldsymbol{\theta})}{Q_X(X)Q_{\boldsymbol{\theta}}(\boldsymbol{\theta})} \\ & \equiv \mathcal{F}_{Q_X Q_{\boldsymbol{\theta}}}.\end{aligned} \tag{3}$$

The VBEM algorithm guarantees finding a local maximum of $\mathcal{F}_{Q_X Q_{\boldsymbol{\theta}}}$ by iteratively performing the following two steps:

- VBE Step: fix $Q_{\boldsymbol{\theta}}$ and update

$$Q_{\boldsymbol{x}_i}(\boldsymbol{x}_i) \propto \exp\Big\{\langle \log P(\boldsymbol{y}_i, \boldsymbol{x}_i|\boldsymbol{\theta})\rangle_{Q_{\boldsymbol{\theta}}}\Big\},$$

- VBM Step: fix Q_X and update

$$Q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \propto P(\boldsymbol{\theta}) \exp\left\{\sum_{i=1}^N \langle \log P(\boldsymbol{y}_i, \boldsymbol{x}_i|\boldsymbol{\theta})\rangle_{Q_{\boldsymbol{x}_i}}\right\},$$

where $\langle\cdot\rangle_Q$ denotes the expectation under a distribution Q. The maximal value of $\mathcal{F}_{Q_X Q_{\boldsymbol{\theta}}}$ gives an approximate log marginal likelihood and the optimal Q_X and $Q_{\boldsymbol{\theta}}$ at the maximum of $\mathcal{F}_{Q_X Q_{\boldsymbol{\theta}}}$ provides an approximate posterior over the latent variables and the model parameters. The quality of approximations is evaluated by the tightness of the lower bound. To simplify notation, we have here and will henceforth assume a given particular model $\mathcal{M}$, even when this is not explicitly stated as the notation in (3).

Next, we consider a more general VB framework than the VBEM approach, which we call LSVB. Rather than exploiting the factorized distribution over the latent variables and the model parameters, in LSVB we integrate out the model parameters in an exact way and then form a lower bound $\mathcal{F}_{Q_X}$ with respect to a factorized approximating distribution $Q_X(X) = \prod_{i=1}^N Q_{\boldsymbol{x}_i}(\boldsymbol{x}_i)$ over samples by using Jensen's inequality:

$$\log P(Y) \geq \int dX Q_X(X) \log \frac{P(Y, X)}{Q_X(X)} \equiv \mathcal{F}_{Q_X}, \tag{4}$$

where $P(Y, X) \equiv \int d\boldsymbol{\theta} P(Y, X|\boldsymbol{\theta})P(\boldsymbol{\theta})$ denotes the complete data marginal likelihood. Although high-dimensional integrals in the marginal likelihood $P(Y)$ for Bayesian inference can be intractable, for many models, the complete data marginal likelihood $P(Y, X)$ is a simple function of sufficient statistics. Thus, no explicit optimization is necessary to compute $P(Y, X)$. Applying the Jensen's inequality to $\mathcal{F}_{Q_X}$ with respect to $Q_{\boldsymbol{\theta}}$, it is easy to see that the lower bound of LSVB is always tighter than the lower bound of VBEM, that is, $\mathcal{F}_{Q_X} \geq \max_{Q_{\boldsymbol{\theta}}} \mathcal{F}_{Q_X Q_{\boldsymbol{\theta}}}$.

Since the lower bound $\mathcal{F}_{Q_X}$ is a concave functional over Q_X, if we set the functional derivative of $\mathcal{F}_{Q_X}$ with respect to $Q_{\boldsymbol{x}_i}$ to zero, we can find the optimal $Q_{\boldsymbol{x}_i}$ at the unique maximum of $\mathcal{F}_{Q_X}$ in the form of

$$Q_{\boldsymbol{x}_i}(\boldsymbol{x}_i) \propto \exp\Big\{\langle \log P(Y, X)\rangle_{\sim Q_{\boldsymbol{x}_i}}\Big\}, \tag{5}$$

where $\langle\cdot\rangle_{\sim Q_{\boldsymbol{x}_i}}$ denotes the expectation under all $Q_{\boldsymbol{x}_{i'}}$ for $i' \neq i$. Analytical solutions for all $\{Q_{\boldsymbol{x}_i}\}$ do not generally exist due to the couplings among them, but, analogously to VBEM, we can locally maximize $\mathcal{F}_{Q_X}$ by iteratively updating $Q_{\boldsymbol{x}_i}$ at one time by fixing all of the others $\{Q_{\boldsymbol{x}_{i'}} : i' \neq i\}$ in a round-robin or random updating schedule. We call this iterative updating procedure the *LSVB algorithm*, which never decreases the lower bound $\mathcal{F}_{Q_X}$ and therefore guarantees finding a local maximum of $\mathcal{F}_{Q_X}$.

In contrast to the VBEM approach, the LSVB approach no longer gives an estimate of the posterior over the model parameters, but it gives better estimates of the model evidence as well as the distribution over the latent variables at the tighter lower bound $\mathcal{F}_{Q_X}$ than the VBEM approach. From this distribution over the latent variables, one can later estimate the posterior over the model parameters. A simple example that is used here is to estimate the posterior over the model parameters by taking a single VBM step with the estimated Q_X.

3 Conjugate-Exponential Family

Consider a class of *exponential family* distributions $P(\boldsymbol{y}_i, \boldsymbol{x}_i|\boldsymbol{\theta})$ for complete data point $(\boldsymbol{y}_i, \boldsymbol{x}_i)$:

$$P(\boldsymbol{y}_i, \boldsymbol{x}_i|\boldsymbol{\theta}) = f(\boldsymbol{y}_i, \boldsymbol{x}_i) g(\boldsymbol{\theta}) \exp\Big\{\boldsymbol{\phi}(\boldsymbol{\theta})^{\mathrm{T}} \mathbf{u}_i(\boldsymbol{y}_i, \boldsymbol{x}_i)\Big\}, \tag{6}$$

where $\mathbf{u}_i(\boldsymbol{y}_i, \boldsymbol{x}_i)$ is a function of the complete data and $\boldsymbol{\phi}$ is called the natural parameters. The functions f and g define the exponential family and g is a constant with respect to $\boldsymbol{y}_i$ and $\boldsymbol{x}_i$ ensuring that the distribution normalizes to one.

We further consider a conjugate prior $P(\boldsymbol{\theta}|\eta^\circ, \boldsymbol{\nu}^\circ)$ over the model parameters given hyperparameters η° and $\boldsymbol{\nu}^\circ$ to the complete data likelihood $P(X, Y|\boldsymbol{\theta}) = \prod_{i=1}^N P(\boldsymbol{x}_i, \boldsymbol{y}_i|\boldsymbol{\theta})$; such priors take the same form as the exponential family:

$$P(\boldsymbol{\theta}|\eta^\circ, \boldsymbol{\nu}^\circ) = h(\eta^\circ, \boldsymbol{\nu}^\circ)^{-1} g(\boldsymbol{\theta})^{\eta^\circ} \exp\Big\{\boldsymbol{\phi}(\boldsymbol{\theta})^{\mathrm{T}} \boldsymbol{\nu}^\circ\Big\}. \tag{7}$$

The normalizing function, $h(\eta^\circ, \boldsymbol{\nu}^\circ) \equiv \int d\boldsymbol{\theta}\, g(\boldsymbol{\theta})^{\eta^\circ} \exp\{\boldsymbol{\phi}(\boldsymbol{\theta})^{\top} \boldsymbol{\nu}^\circ\}$, is known for many standard conjugate priors. This means that we can analytically integrate out model parameters in models using, e.g., Dirichlet, Gaussian, Wishart, Gamma, Beta, and Poisson distributions, and many combinations thereof.

Especially, the exponential family distribution in (6) with the conjugate prior in (7) is called the *conjugate-exponential family* [19], which includes many practical latent variable models such as mixtures of Gaussians, mixtures of Multinomials, mixtures of Bernoullis, mixtures of factor analyzers, state-space models, hidden Markov models, linear dynamical systems, and some kinds of graphical models. In particular, the conjugate-exponential family has the posterior $P(\boldsymbol{\theta}|Y, X, \eta^\circ, \boldsymbol{\nu}^\circ)$ over the model parameters after observing the complete data set (Y, X) in the same form as the prior in (7), that is, $P(\boldsymbol{\theta}|Y, X, \eta^\circ, \boldsymbol{\nu}^\circ) = P(\boldsymbol{\theta}|\eta, \boldsymbol{\nu})$ with $\eta = N + \eta^\circ$ and $\boldsymbol{\nu} = \sum_{i=1}^N \mathbf{u}_i(\boldsymbol{y}_i, \boldsymbol{x}_i) + \boldsymbol{\nu}^\circ$. We will use the term *prior hyperparameters* to refer to η° and $\boldsymbol{\nu}^\circ$, and the term *posterior hyperparameters* to refer to η and $\boldsymbol{\nu}$.

3.1 LSVB

The conjugate-exponential family has the complete data marginal likelihood composed of analytically known functions in the form of

$$\begin{aligned} P(Y,X|\eta^\circ,\boldsymbol{\nu}^\circ) &= \int d\boldsymbol{\theta} P(Y,X|\boldsymbol{\theta})P(\boldsymbol{\theta}|\eta^\circ,\boldsymbol{\nu}^\circ) \\ &= \frac{h(\eta,\boldsymbol{\nu})}{h(\eta^\circ,\boldsymbol{\nu}^\circ)}\prod_{i=1}^N f(\boldsymbol{y}_i,\boldsymbol{x}_i), \end{aligned} \tag{8}$$

where $\eta = N + \eta^\circ$ and $\boldsymbol{\nu} = \sum_{i=1}^N \mathbf{u}_i(\boldsymbol{y}_i,\boldsymbol{x}_i) + \boldsymbol{\nu}^\circ$. Plugging this $P(Y,X|\eta^\circ,\boldsymbol{\nu}^\circ)$ into $\mathcal{F}_{Q_X}$ in (4), the lower bound of LSVB can be formulated by

$$\mathcal{F}_{Q_X} = R_{Q_X} + \langle \log h(\eta,\boldsymbol{\nu})\rangle_{Q_X}, \tag{9}$$

with

$$R_{Q_X} \equiv \sum_{i=1}^N \langle \log f(\boldsymbol{y}_i,\boldsymbol{x}_i)\rangle_{Q_{\boldsymbol{x}_i}} - \log h(\eta^\circ,\boldsymbol{\nu}^\circ) + \sum_{i=1}^N \mathcal{H}(Q_{\boldsymbol{x}_i}),$$

where $\mathcal{H}$ denotes the entropy defined by

$$\mathcal{H}(Q) \equiv -\int dt Q(t)\log Q(t).$$

The LSVB algorithm maximizes this $\mathcal{F}_{Q_X}$ by iteratively updating

$$Q_{\boldsymbol{x}_i}(\boldsymbol{x}_i) \propto f(\boldsymbol{y}_i,\boldsymbol{x}_i)\exp\Big\{\langle \log h(\eta,\boldsymbol{\nu})\rangle_{\sim Q_{\boldsymbol{x}_i}}\Big\}. \tag{10}$$

To see the relation to VBEM, let us consider the first-order Taylor's expansion of $\log h$ around $\tilde{\boldsymbol{\nu}}$ given by

$$\log \tilde{h}(\eta,\boldsymbol{\nu};\tilde{\boldsymbol{\nu}}) \equiv \log h(\eta,\tilde{\boldsymbol{\nu}}) + (\boldsymbol{\nu}-\tilde{\boldsymbol{\nu}})^{\mathrm{T}}\mathbf{m}_{\boldsymbol{\phi}}(\tilde{\boldsymbol{\nu}}), \tag{11}$$

where $\mathbf{m}_{\boldsymbol{\phi}}(\tilde{\boldsymbol{\nu}}) \equiv \langle \boldsymbol{\phi}(\boldsymbol{\theta})\rangle_{P(\boldsymbol{\theta}|\eta,\tilde{\boldsymbol{\nu}})}$ is the gradient vector of $\log h$ evaluated at $\boldsymbol{\nu} = \tilde{\boldsymbol{\nu}}$. Incorporating the inequality $\log h(\eta,\boldsymbol{\nu}) \geq \log \tilde{h}(\eta,\boldsymbol{\nu};\tilde{\boldsymbol{\nu}})$ for all $\tilde{\boldsymbol{\nu}}$ from the convexity of $\log h$, we can form a first-order lower bound $\mathcal{F}_{Q_X,\tilde{\boldsymbol{\nu}}}$ of $\mathcal{F}_{Q_X}$:

$$\mathcal{F}_{Q_X} \geq R_{Q_X} + \log \tilde{h}\Big(\eta, \langle \boldsymbol{\nu}\rangle_{Q_X};\tilde{\boldsymbol{\nu}}\Big) \equiv \mathcal{F}_{Q_X,\tilde{\boldsymbol{\nu}}}. \tag{12}$$

Theorem 1. *For the conjugate-exponential family, the VBEM algorithm maximizes the first-order lower bound $\mathcal{F}_{Q_X,\tilde{\boldsymbol{\nu}}}$, applying the following two steps:*

- *VBE Step: fix $\tilde{\boldsymbol{\nu}}$ and find $Q_{\boldsymbol{x}_i} \leftarrow \arg\max_{Q_{\boldsymbol{x}_i}} \mathcal{F}_{Q_X,\tilde{\boldsymbol{\nu}}}$,*
- *VBM Step: fix Q_X and find $\tilde{\boldsymbol{\nu}} \leftarrow \arg\max_{\tilde{\boldsymbol{\nu}}} \mathcal{F}_{Q_X,\tilde{\boldsymbol{\nu}}}$.*

(The proof is given in Appendix A.)

From Theorem 1, we can find that the VBEM algorithm is essentially an Expectation-Maximization algorithm [22], a standard method for maximum likelihood parameter estimation, at the level of posterior hyperparameters. Therefore, the VBEM algorithm can have the following problems: First, maximizing over the posterior hyperparameters, the VBEM algorithm can fall into spurious local maxima and suffer from overfitting the posterior hyperparameters. Second, fixing the posterior hyperparameters in the VBE step prevents the latent variables over samples from interacting with each other, which can make the algorithm ignore some correlations as well as converge slowly.

Integrating out the model parameters, LSVB does not involve the problems of VBEM above, but, in practice, it requires the difficult expectation of the nonlinear function $\log h$ to be approximated. For easy and wide applicability, we next introduce a new VB approximate inference method as a practical implementation of LSVB.

3.2 First-Order LSVB

Generally, the difficult expectation $\langle \log h(\eta,\boldsymbol{\nu})\rangle_{Q_X}$ under a distribution Q_X can be approximated by replacing the nonlinear function $\log h$ with the first-order (linear) approximate $\log \tilde{h}$ at $\tilde{\boldsymbol{\nu}} = \langle \nu\rangle_{Q_X}$, which reduces to

$$\langle \log h(\eta,\boldsymbol{\nu})\rangle_{Q_X} \approx \log h\Big(\eta, \langle \boldsymbol{\nu}\rangle_{Q_X}\Big). \tag{13}$$

This standard approximation for the difficult expectation is quite simple as neither the gradient nor the Hessian of $\log h$ is required and it can be directly incorporated in LSVB.

Incorporating $\langle \log h(\eta,\boldsymbol{\nu})\rangle_{\sim Q_{\boldsymbol{x}_i}} \approx \log h(\eta, \langle \boldsymbol{\nu}\rangle_{\sim Q_{\boldsymbol{x}_i}})$ in (10) makes the first-order approximate $Q_{\boldsymbol{x}_i}$ to be in the form of

$$Q_{\boldsymbol{x}_i}(\boldsymbol{x}_i) \propto f(\boldsymbol{y}_i,\boldsymbol{x}_i)h\Big(\eta, \mathbf{u}_i(\boldsymbol{y}_i,\boldsymbol{x}_i) + \langle \boldsymbol{\nu}^{\neg i}\rangle_{\sim Q_{\boldsymbol{x}_i}}\Big), \tag{14}$$

where $\boldsymbol{\nu}^{\neg i} = \sum_{i'=1,\neq i}^N \mathbf{u}_{i'}(\boldsymbol{y}_{i'},\boldsymbol{x}_{i'}) + \boldsymbol{\nu}^\circ$. We use the notation $\neg i$ for excluding the ith sample. We can iteratively update the first-order approximate $Q_{\boldsymbol{x}_i}$ in (14) instead of the exact $Q_{\boldsymbol{x}_i}$ in (10). We call this approximate iterative procedure to infer the distribution over the latent variables *FoLSVB* algorithm. The FoLSVB algorithm has some advantages over the VBEM algorithm and requires the same computational complexity. First, the FoLSVB algorithm directly approximates the distribution over the latent variables obtained at the tighter lower bound $\mathcal{F}_{Q_X}$ than $\mathcal{F}_{Q_X,\tilde{\boldsymbol{\nu}}}$ of VBEM. This means that it can find a better distribution over the latent variables than the VBEM algorithm. Second, the FoLSVB algorithm can alleviate the problem of overfitting the posterior hyperparameters since no maximization over the posterior hyperparameters is associated with it. Third, direct interactions among the latent variables over samples can make the FoLSVB algorithm capture more correlations and converge faster than the VBEM algorithm. A weakness of the FoLSVB algorithm is a lack of theoretical convergence guarantees, but it turns out that it converges very well in practice.

The FoLSVB algorithm gives an estimate of the distribution over the latent variables, but, in contrast to the VBEM algorithm, does not explicitly give an estimate of the model evidence. However, in the FoLSVB framework, the optimal lower bound $\mathcal{F}_{Q_X}$ can be directly approximated by incorporating $\langle \log h(\eta,\boldsymbol{\nu})\rangle_{Q_X} \approx \log h(\eta, \langle \boldsymbol{\nu}\rangle_{Q_X})$ with Q_X estimated by the FoLSVB algorithm such as

$$\mathcal{F}_{Q_X} \approx R_{Q_X} + \log h\Big(\eta, \langle \boldsymbol{\nu}\rangle_{Q_X}\Big) \equiv \widetilde{\mathcal{F}}_{Q_X}. \tag{15}$$

Since the first-order lower bound $\mathcal{F}_{Q_X,\tilde{\boldsymbol{\nu}}}$ of VBEM becomes tight after the VBM step, it reduces to the same form of $\widetilde{\mathcal{F}}_{Q_X}$, that is, $\widetilde{\mathcal{F}}_{Q_X} = \max_{\tilde{\boldsymbol{\nu}}} \mathcal{F}_{Q_X,\tilde{\boldsymbol{\nu}}}$. Therefore, both FoLSVB and VBEM estimate the log evidence by the same form of $\widetilde{\mathcal{F}}_{Q_X}$ but use different estimates of Q_X to do so. Note that, in the FoLSVB framework, we estimate $\widetilde{\mathcal{F}}_{Q_X}$ at the last stage after estimating Q_X by the FoLSVB algorithm rather than explicitly maximizing $\widetilde{\mathcal{F}}_{Q_X}$. This means that, in principle, the VBEM algorithm directly maximizing $\widetilde{\mathcal{F}}_{Q_X}$ can find a tighter lower bound than FoLSVB at the global maximum. However, we will see that, in practice, the VBEM algorithm is fraught with local maxima, which lead to a looser lower bound than FoLSVB.

Recently, the CV approximation was proposed for specific Dirichlet-Multinomial models called Latent Dirichlet Allocation [20] and Hierarchical Dirichlet Process [21]. Our LSVB framework here generalizes their CV approximation to a more general class of conjugate-exponential models. Especially, it can be shown that their practical CV approximation for Dirichlet-Multinomial models that incorporates Gaussian approximation technique is a special case of the second-order extension of FoLSVB in which the first-order approximation of $\log h$ in FoLSVB is replaced by the second-order approximation. The proof is straightforward but is long, so we will not give it here.

4 Mixture of Gaussians

Finite mixture [23], [24] is a latent variable model which provides a natural framework for cluster analysis and density estimation of an unknown distribution. For a finite mixture, the number of mixture components, K, represents the model $\mathcal{M}$. Also, it is convenient to use a K-dimensional indicator variable to represent the latent variables such as $\boldsymbol{x}_i = (x_{i1}, \dots, x_{iK})$ with only a single element taking on the value one and all other elements being zero.

We consider here the mixture of Gaussians (MoG), a standard mixture model for continuous observed data that has Gaussian densities as its mixture components. Consider a D-dimensional continuous data $\boldsymbol{y}_i$. For MoG, the joint distribution $P(\boldsymbol{y}_i, \boldsymbol{x}_i|\boldsymbol{\theta})$ given model parameters $\boldsymbol{\theta} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\lambda}_k\}_{k=1}^K$ can be written as

$$P(\boldsymbol{y}_i, \boldsymbol{x}_i|\boldsymbol{\theta}) = \prod_{k=1}^K [\pi_k \mathrm{N}(\boldsymbol{y}_i|\boldsymbol{\mu}_k, \boldsymbol{\lambda}_k)]^{x_{ik}}, \tag{16}$$

where the mixing coefficient π_k satisfies $0 \le \pi_k \le 1$ and $\sum_{k=1}^K \pi_k = 1$. The standard Gaussian density $\mathrm{N}(\cdot|\boldsymbol{\mu}_k, \boldsymbol{\lambda}_k)$ with mean vector $\boldsymbol{\mu}_k$ and precision matrix $\boldsymbol{\lambda}_k$ represents the kth mixture component. The Dirichlet(D) prior on $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)$ and the Normal(N)-Wishart(W) prior on $(\boldsymbol{\mu}_k, \boldsymbol{\lambda}_k)$ constitute a conjugate prior over the model parameters of MoG: given prior hyperparameters $\varphi^\circ = \{\alpha_k^\circ, \tau_k^\circ, r_k^\circ, \boldsymbol{\xi}_k^\circ, \boldsymbol{B}_k^\circ\}_{k=1}^K$,

$$P(\boldsymbol{\theta}|\boldsymbol{\varphi}^\circ) = \mathrm{D}(\boldsymbol{\pi}|\boldsymbol{\alpha}^\circ) \prod_{k=1}^K \mathrm{N}\big(\boldsymbol{\mu}_k|\boldsymbol{\xi}_k^\circ, \tau_k^\circ \boldsymbol{\lambda}_k\big) \mathrm{W}\big(\boldsymbol{\lambda}_k|r_k^\circ, \boldsymbol{B}_k^\circ\big), \tag{17}$$

where $\boldsymbol{\alpha}^\circ = (\alpha_1^\circ, \dots, \alpha_K^\circ)$. The forms of standard distributions in (17) are given in Appendix B. Also, the normalizing function h can be written in the form of

$$h(\boldsymbol{\varphi}^\circ) = \frac{2^{\frac{DK}{2}} \pi^{\frac{D(D+1)K}{4}}}{\Gamma\Big(\sum_{k'=1}^K \alpha_{k'}^\circ\Big)} \prod_{k=1}^K \frac{\Gamma(\alpha_k^\circ) \prod_{l=1}^D \Gamma\big(r_k^\circ + \frac{1-l}{2}\big)}{(\tau_k^\circ)^{\frac{D}{2}} \big|\boldsymbol{B}_k^\circ\big|^{r_k{}^\circ}}, \tag{18}$$

where $\Gamma(\cdot)$ denotes the standard gamma function.

4.1 FoLSVB

For MoG, the first-order approximate $Q_{\boldsymbol{x}_i}$ in (14) can be simplified in the form of weighted Student distributions with the hyperparameters $\tilde{\varphi}^{\neg i} = \{\tilde{\alpha}_k^{\neg i}, \tilde{\tau}_k^{\neg i}, \tilde{r}_k^{\neg i}, \tilde{\boldsymbol{\xi}}_k^{\neg i}, \tilde{\boldsymbol{B}}_k^{\neg i}\}_{k=1}^K$:

$$Q_{\boldsymbol{x}_i}(x_{ik} = 1) \equiv \gamma_{ik} \propto \tilde{\alpha}_k^{\neg i} \mathrm{S}\big(\boldsymbol{y}_i\big|\tilde{\boldsymbol{\xi}}_k^{\neg i}, \tilde{\boldsymbol{\Lambda}}_k^{\neg i}, 2\tilde{r}_k^{\neg i} - D + 1\big) \tag{19}$$

with $\tilde{\boldsymbol{\Lambda}}_k^{\neg i} = \frac{(\tilde{r}_k^{\neg i} - \frac{D}{2} + \frac{1}{2})\tilde{\tau}_k^{\neg i}}{\tilde{\tau}_k^{\neg i} + 1} (\tilde{\boldsymbol{B}}_k^{\neg i})^{-1}$, where S denotes the standard Student distribution (see Appendix B). The hyperparameter $\tilde{\varphi}^{\neg i}$ excluding the ith sample is given by

$$\begin{aligned} \tilde{\alpha}_k^{\neg i} &= \alpha_k^\circ + \langle n_k^{\neg i} \rangle, \quad \tilde{\tau}_k^{\neg i} = \tau_k^\circ + \langle n_k^{\neg i} \rangle, \\ \tilde{r}_k^{\neg i} &= r_k^\circ + \frac{1}{2} \langle n_k^{\neg i} \rangle, \tilde{\boldsymbol{\xi}}_k^{\neg i} = \big(\tilde{\tau}_k^{\neg i}\big)^{-1} \big(\tau_k^\circ \boldsymbol{\xi}_k^\circ + \langle \boldsymbol{\rho}_k^{\neg i} \rangle\big), \\ \tilde{\boldsymbol{B}}_k^{\neg i} &= \boldsymbol{B}_k^\circ + \frac{1}{2} \big(\tau_k^\circ \boldsymbol{\xi}_k^\circ \boldsymbol{\xi}_k^{\circ \mathrm{T}} - \tilde{\tau}_k^{\neg i} \tilde{\boldsymbol{\xi}}_k^{\neg i} \tilde{\boldsymbol{\xi}}_k^{\neg i \mathrm{T}} + \langle \boldsymbol{W}_k^{\neg i} \rangle\big), \end{aligned} \tag{20}$$

where $\langle n_k^{\neg i} \rangle = \sum_{i'=1, \neq i}^N \gamma_{i'k}$, $\langle \boldsymbol{\rho}_k^{\neg i} \rangle = \sum_{i'=1, \neq i}^N \gamma_{i'k} \boldsymbol{y}_{i'}$, and $\langle \boldsymbol{W}_k^{\neg i} \rangle = \sum_{i'=1, \neq i}^N \gamma_{i'k} \boldsymbol{y}_{i'} \boldsymbol{y}_{i'}^\top$. Note that γ_{ik} should be properly normalized to be $\sum_{k=1}^K \gamma_{ik} = 1$. Since there is no explicit maximization over the posterior hyperparameters, the FoLSVB algorithm is quite simple in that only $Q_{\boldsymbol{x}_i}$ in (19) are repeatedly updated one by one, while the others are fixed.

In practice, the hyperparameters $\tilde{\varphi}^{\neg i}$ at each updating can be efficiently computed by taking out terms associated with the ith sample from $\tilde{\varphi} = \{\tilde{\alpha}_k, \tilde{\tau}_k, \tilde{r}_k, \tilde{\boldsymbol{\xi}}_k, \tilde{\boldsymbol{B}}_k\}_{k=1}^K$ given by

$$\begin{aligned} \tilde{\alpha}_k &= \alpha_k^\circ + \langle n_k \rangle, \quad \tilde{\tau}_k = \tau_k^\circ + \langle n_k \rangle, \\ \tilde{r}_k &= r_k^\circ + \frac{1}{2} \langle n_k \rangle, \tilde{\boldsymbol{\xi}}_k = (\tilde{\tau}_k)^{-1} \big(\tau_k^\circ \boldsymbol{\xi}_k^\circ + \langle \boldsymbol{\rho}_k \rangle\big), \\ \tilde{\boldsymbol{B}}_k &= \boldsymbol{B}_k^\circ + \frac{1}{2} \big(\tau_k^\circ \boldsymbol{\xi}_k^\circ \boldsymbol{\xi}_k^{\circ \mathrm{T}} - \tilde{\tau}_k \tilde{\boldsymbol{\xi}}_k \tilde{\boldsymbol{\xi}}_k^{\mathrm{T}} + \langle \boldsymbol{W}_k \rangle\big), \end{aligned} \tag{21}$$

where $\langle n_k \rangle = \sum_{i=1}^N \gamma_{ik}$, $\langle \boldsymbol{\rho}_k \rangle = \sum_{i=1}^N \gamma_{ik} \boldsymbol{y}_i$, and $\langle \boldsymbol{W}_k \rangle = \sum_{i=1}^N \gamma_{ik} \boldsymbol{y}_i \boldsymbol{y}_i^\top$. In Table 1, we present this efficient FoLSVB algorithm for MoG. Also, we can efficiently compute the inverse of $\tilde{\boldsymbol{B}}_k^{\neg i}$ required in $\tilde{\boldsymbol{\Lambda}}_k^{\neg i}$ by using one-rank matrix inverse formula such that $(\tilde{\boldsymbol{B}}_k^{\neg i})^{-1} = \tilde{\boldsymbol{B}}_k^{-1} - C_k^{-1} \tilde{\boldsymbol{B}}_k^{-1} (\boldsymbol{y}_i - \tilde{\boldsymbol{\xi}}_k)(\boldsymbol{y}_i - \tilde{\boldsymbol{\xi}}_k)^\top \tilde{\boldsymbol{B}}_k^{-1}$, where $C_k = (\boldsymbol{y}_i - \tilde{\boldsymbol{\xi}}_k)^\top \tilde{\boldsymbol{B}}_k^{-1} (\boldsymbol{y}_i - \tilde{\boldsymbol{\xi}}_k) - \frac{2(\tilde{\tau}_k - \gamma_{ik})}{\tilde{\tau}_k \gamma_{ik}}$. The cost for these efficient computations is only small additional storages for $\tilde{\varphi}$ and $\tilde{\boldsymbol{B}}_k^{-1}$ to always keep them up-to-date after updating $Q_{\boldsymbol{x}_i}$.

TABLE 1
FoLSVB Algorithm for the Mixture of Gaussians

```
(a) Initialize all $\{\gamma_{ik}\}$, $i = 1, \dots, N$ and $k = 1, \dots, K$.
(b) Compute $\tilde{\boldsymbol{\varphi}}$ given by (21).
(c) Choose $i \in \{1, \dots, N\}$ and do
    (i) Compute $\tilde{\boldsymbol{\varphi}}^{\neg i}$ by removing the $i$-th sample from $\tilde{\boldsymbol{\varphi}}$:
        $\tilde{\alpha}_k^{\neg i} = \tilde{\alpha}_k - \gamma_{ik}$, $\tilde{\tau}_k^{\neg i} = \tilde{\tau}_k - \gamma_{ik}$, $\tilde{r}_k^{\neg i} = \tilde{r}_k - \frac{1}{2}\gamma_{ik}$,
        $\tilde{\boldsymbol{\xi}}_k^{\neg i} = (\tilde{\tau}_k - \gamma_{ik})^{-1} (\tilde{\tau}_k \tilde{\boldsymbol{\xi}}_k - \gamma_{ik} \boldsymbol{y}_i)$,
        $\tilde{\boldsymbol{B}}_k^{\neg i} = \tilde{\boldsymbol{B}}_k - \frac{\tilde{\tau}_k \gamma_{ik}}{2(\tilde{\tau}_k - \gamma_{ik})} (\boldsymbol{y}_i - \tilde{\boldsymbol{\xi}}_k)(\boldsymbol{y}_i - \tilde{\boldsymbol{\xi}}_k)^\top$.
    (ii) Update $\gamma_{ik}$ by (19) for all $k = 1, \dots, K$.
    (iii) Update $\tilde{\boldsymbol{\varphi}}$ by incorporating the $i$-th sample from $\tilde{\boldsymbol{\varphi}}^{\neg i}$:
        $\tilde{\alpha}_k = \tilde{\alpha}_k^{\neg i} + \gamma_{ik}$, $\tilde{\tau}_k = \tilde{\tau}_k^{\neg i} + \gamma_{ik}$, $\tilde{r}_k = \tilde{r}_k^{\neg i} + \frac{1}{2}\gamma_{ik}$,
        $\tilde{\boldsymbol{\xi}}_k = (\tilde{\tau}_k^{\neg i} + \gamma_{ik})^{-1} (\tilde{\tau}_k^{\neg i} \tilde{\boldsymbol{\xi}}_k^{\neg i} + \gamma_{ik} \boldsymbol{y}_i)$,
        $\tilde{\boldsymbol{B}}_k = \tilde{\boldsymbol{B}}_k^{\neg i} + \frac{\tilde{\tau}_k^{\neg i} \gamma_{ik}}{2(\tilde{\tau}_k^{\neg i} + \gamma_{ik})} (\boldsymbol{y}_i - \tilde{\boldsymbol{\xi}}_k^{\neg i})(\boldsymbol{y}_i - \tilde{\boldsymbol{\xi}}_k^{\neg i})^\top$.
(d) Repeat (c) until there are no changes in all $\{\gamma_{ik}\}$.
```

After converging, the first-order lower bound estimates the log evidence in the form of

$$\begin{aligned} \widetilde{\mathcal{F}}_{Q_X} = &- \frac{ND}{2} \log 2\pi - \log h(\boldsymbol{\varphi}^\circ) \\ &+ \log h(\tilde{\varphi}) - \sum_{i=1}^N \sum_{k=1}^K \gamma_{ik} \log \gamma_{ik}, \end{aligned} \tag{22}$$

where the posterior hyperparameters $\tilde{\varphi}$ incorporating all samples are given in (21).

4.2 Numerical Results

We used the common prior hyperparameters for all components with $\alpha_k^\circ = 1$, $r_k^\circ = 1 + 0.5D$, and $\boldsymbol{\xi}_k^\circ = \mathrm{sample\ mean}$. Especially, $\boldsymbol{B}_k^\circ$ was set for $\langle \boldsymbol{\lambda}_k \rangle$ under the prior to be $(0.3\sigma_{\max})^{-2} \boldsymbol{I}_D$ and then τ_k° was set for the precision of $\boldsymbol{\mu}_k$ to be $(10\sigma_{\max})^{-2} \boldsymbol{I}_D$, where $\sigma_{\max}$ denotes the maximum standard deviation of data set among dimensions. These prior hyperparameters represent that each component covers a subregion of data set but places at a fairly uncertain location in the range of data set. We note that a prior should not depend on a data set in principle, but it is useful to set the prior over the model paratmeters using a data set in practice when we do not have any information about the model parameters a priori.

Both the FoLSVB and the VBEM algorithms were started with the same initial $\{\gamma_{ik}\}$ estimated by $\gamma_{ik} \propto \mathrm{N}(\boldsymbol{y}_i|\boldsymbol{c}_k, (0.3\sigma_{\max})^{-2} \boldsymbol{I}_D)$ with the center $\boldsymbol{c}_k$ of the kth cluster found by k-means algorithm. Also, we considered the algorithms to be strictly converged when the successive changes in all $\{\gamma_{ik}\}$ were very small such that $\frac{1}{NK} \sum_{i=1}^N \sum_{k=1}^K |\gamma_{ik}^{(t)} - \gamma_{ik}^{(t-1)}| < 10^{-9}$, where t denotes iterations. A single iteration means that all $\{\gamma_{ik}\}$ are updated once for the FoLSVB algorithm and the VBE and VBM steps are performed once for the VBEM algorithm.

4.2.1 *Synthetic Data Sets*

To see the basic properties of the algorithms, we first used a 1D toy data set of 20 data points shown in Fig. 1a, which were generated

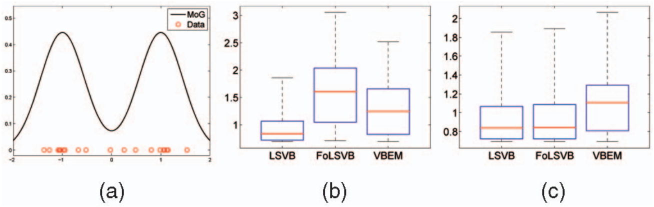

Fig. 1. Results based on 20 trials with different random data sets of 20 data points, generated from the mixture of two Gaussians. All inferences were performed on the model with $K = 2$. (a) Sampling distribution and a single case of random data set. (b) Gap of the lower bound from the true log model evidence, i.e., $\log P(Y) - \mathcal{F}_{Q_X}$ for LSVB and $\log P(Y) - \widetilde{\mathcal{F}}_{Q_X}$ for FoLSVB and VBEM. (c) KL divergence of the true distribution over the latent variables from the approximate distribution.

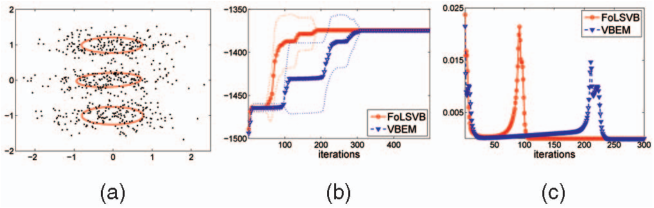

Fig. 2. Demonstration of the convergence speed, based on a synthetic data set of 600 data points generated from the mixture of three Gaussians. Algorithms were performed on the model with $K = 3$. (a) Final Gaussian components by both the FoLSVB and the VBEM algorithms. (b) Average intermediate $\widetilde{\mathcal{F}}_{Q_X}$ at each iteration, annotated by one standard derivation, over 20 trials with different initial $\{\gamma_{ik}\}$ by k-means algorithm. (c) Successive changes in Q_X in terms of $\frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K}|\gamma_{ik}^{(t)} - \gamma_{ik}^{(t-1)}|$ on a single trial.

from the mixture of two Gaussians with $\pi = (0.5, 0.5)$, $\mu_1 = 1$, $\mu_2 = -1$, and $\lambda_1 = \lambda_2 = 5$. For this small data set, we can perform the exact Bayesian inference and the exact LSVB algorithm. Fig. 1 compares the results between approximate inference and true inference performed on the model with $K = 2$. Obviously, LSVB achieves the best estimates of the model evidence as well as the distribution over the latent variables. For this data set, the VBEM algorithm is considered to be finding a good solution without falling into a local maximum, so it finds a tighter lower bound than FoLSVB (Fig. 1b). Directly approximating the distribution over the latent variables of LSVB, FoLSVB, however, gives better distribution over the latent variables than VBEM in terms of KL divergence to the true distribution, showing a very accurate approximation of LSVB (Fig. 1c).

In order to see the convergence speed of the algorithms, we next prepared a well-clustered data set of 600 data points, shown in Fig. 2a, which were generated from the mixture of three Gaussians with $\pi = (1/3, 1/3, 1/3)$, $\boldsymbol{\mu}_1 = (0, 1)$, $\boldsymbol{\mu}_2 = (0, 0)$, $\boldsymbol{\mu}_3 = (0, -1)$, and $\boldsymbol{\lambda}_k = \mathrm{diag}(1.3, 20)$ for $k = 1, \ldots, 3$. Fig. 2 shows the results performed on the model with $K = 3$. Starting from the same initial $\{\gamma_{ik}\}$ by k-means algorithm, the FoLSVB algorithm converges very well and much faster than the VBEM algorithm, while both algorithms find almost the same solution. In Fig. 2c, FoLSVB meets the convergence threshold, 10^{-9}, at 124 iterations, but VBEM requires 262 iterations to do so.

4.2.2 Real-World Data Sets

Next, we considered six real-world data sets in different dimensions, called *Enzyme*, *Old Faithful*, *Iris*, *Glass*, *Wine*, and *Ionosphere*, which have often been used to demonstrate inference algorithms in the pattern recognition, statistics, and machine learning literatures [6], [24], [25], [26]. The Ionosphere data were originally 34D, but the dimension was reduced to 26 by using the standard Principle Component Analysis (PCA) dimensionality reduction technique [6]. We further standardized these data sets along each dimension, except for the 1D Enzyme data set, so that mean and variance become zero and one, respectively. The prior hyperparameters were set by using the standardized data set. The results are shown in Fig. 3. For low-dimensional data sets up to Iris ($D \leq 3$), both FoLSVB and VBEM find the same model as the best while VBEM gives a similar or tighter lower bound for the models with more components than the best model. However, for high-dimensional data sets with $D \geq 9$, FoLSVB finds a significantly tighter lower bound than VBEM. Furthermore, the variance over 30 trials for FoLSVB is smaller than VBEM, showing a lower sensitivity to initial conditions. We can see that VBEM is very vulnerable to the high-dimensional data sets. In most cases, VBEM falls into a different spurious local maximum, depending on initial conditions.

To compare the convergence speed, we performed both the FoLSVB and the VBEM algorithms 50 times with different initial $\{\gamma_{ik}\}$ for the cases where they found a similar model evidence in Fig. 3. Differently from the previous examples, we initialized $\{\gamma_{ik}\}$ by randomly choosing $\boldsymbol{c}_k$ among the data set rather than estimating by k-means algorithm. Next, we chose trials over these 50 trials such that both algorithms found almost the same model evidence as well as the posterior over the model parameters. Based on final chosen trials, the numbers of iterations reached at the convergence threshold, 10^{-9}, are shown in Table 2. In FoLSVB, direct interactions among the latent variables over samples can make the information propagate more rapidly than VBEM so we can see that FoLSVB converges much faster than VBEM when they find a near same solution.

Addressing the problem of data dimension, we last used a 64D numeral digit "1" data set of 500 data points, which is a part of optical handwritten digit data sets in the UCI data repository [25]. We reduced the dimension by 5, 10, 20, and 30 by the PCA technique as before. The results are shown in Fig. 4. The more the dimension of data increases, the worse the local maxima problem becomes with VBEM. In all cases with $D =$ 10, 20, and 30, FoLSVB gives a much tighter lower bound than VBEM.

Enzyme $N = 245$ $D = 1$	Old faithful $N = 222$ $D = 2$	Iris $N = 150$ $D = 3$	Glass $N = 214$ $D = 9$	Wine $N = 178$ $D = 13$	Ionosphere $N = 351$ $D = 26$

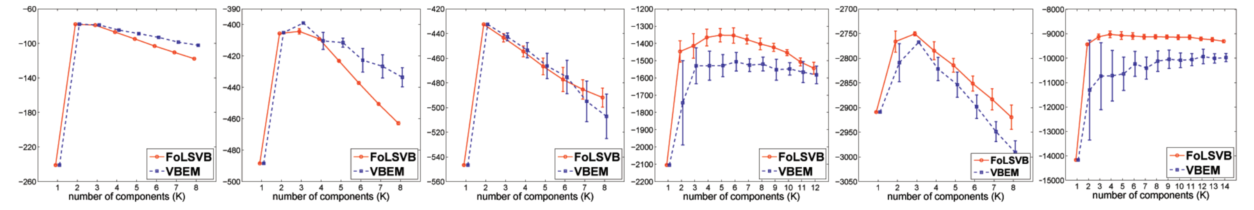

Fig. 3. Lower bounds, $\widetilde{\mathcal{F}}_{Q_X}$, on the real-world data sets. The results are averaged and annotated by one standard deviation over 30 trials with different initial $\{\gamma_{ik}\}$ by k-means algorithm.

TABLE 2
Number of Iterations

Data set	FoLSVB	VBEM
Enzyme ($K = 3$)	187.52 ± 13.90	382.34 ± 92.64
Old Faithful ($K = 2$)	133.89 ± 29.48	365.62 ± 154.32
Iris ($K = 2$)	8.60 ± 7.90	17.02 ± 22.31
Wine ($K = 3$)	20.89 ± 9.45	36.34 ± 21.40

5 Mixture of Bernoullis

We considered MoG as a finite mixture model for continuous observed data. Next, we give another example of the finite mixture model for discrete binary observed data, which has Bernoulli distributions as its mixture components [23]. Suppose an i.i.d. data set $Y = \{\boldsymbol{y}_i\}_{i=1}^N$ of D-dimensional binary data point $\boldsymbol{y}_i = (y_{ij})_{j=1}^D$ with $y_{ij} \in \{0, 1\}$. The mixture of Bernoullis (MoB) has the joint distribution of complete data point governed by mixing coefficients π_k and mean parameters $\boldsymbol{\mu}_k = (\mu_{kj})_{j=1}^D$ of the kth Bernoulli component: Given model parameters $\boldsymbol{\theta} = \{\pi_k, \boldsymbol{\mu}_k\}_{k=1}^K$,

$$P(\boldsymbol{y}_i, \boldsymbol{x}_i|\boldsymbol{\theta}) = \prod_{k=1}^K \left[\pi_k \prod_{j=1}^D \mathrm{Bern}(y_{ij}|\mu_{kj})\right]^{x_{ik}}, \tag{23}$$

where $\mathrm{Bern}(y_{ij}|\mu_{kj}) \equiv \mu_{kj}^{y_{ij}}(1 - \mu_{kj})^{1-y_{ij}}$ denotes the standard Bernoulli distribution. The Dirichlet distribution and Beta distributions (see Appendix B) give a conjugate prior over the model parameters of MoB: Given prior hyperparameters $\varphi^\circ = \{\alpha_k^\circ, (\beta_{kj,(1)}^\circ)_{j=1}^D, (\beta_{kj,(2)}^\circ)_{j=1}^D\}_{k=1}^K$,

$$P(\boldsymbol{\theta}|\varphi^\circ) = \mathrm{D}(\boldsymbol{\pi}|\boldsymbol{\alpha}^\circ) \prod_{k=1}^K \prod_{j=1}^D \mathrm{Beta}\Big(\mu_{kj}|\beta_{kj,(1)}^\circ, \beta_{kj,(2)}^\circ\Big) \tag{24}$$

and the normalizing function is given by

$$h(\varphi^\circ) = \frac{1}{\Gamma\Big(\sum_{k'=1}^K \alpha_{k'}^\circ\Big)} \prod_{k=1}^K \Gamma(\alpha_k^\circ) \prod_{j=1}^D \frac{\Gamma\Big(\beta_{kj,(1)}^\circ\Big)\Gamma\Big(\beta_{kj,(2)}^\circ\Big)}{\Gamma\Big(\beta_{kj,(1)}^\circ + \beta_{kj,(2)}^\circ\Big)}.$$

5.1 FoLSVB

The FoLSVB algorithm iteratively updates $\{\gamma_{ik}\}_{k=1}^K$ given all of the others in the form of weighted Bernoulli distributions with $\tilde{\varphi}^{\neg i} = \{\tilde{\alpha}_k^{\neg i}, (\tilde{\beta}_{kj,(1)}^{\neg i})_{j=1}^D, (\tilde{\beta}_{kj,(2)}^{\neg i})_{j=1}^D\}_{k=1}^K$:

$$\gamma_{ik} \propto \tilde{\alpha}_k^{\neg i} \prod_{j=1}^D \mathrm{Bern}\Big(y_{ij}|\tilde{\beta}_{kj,(1)}^{\neg i} / \Big(\tilde{\beta}_{kj,(1)}^{\neg i} + \tilde{\beta}_{kj,(2)}^{\neg i}\Big)\Big), \tag{25}$$

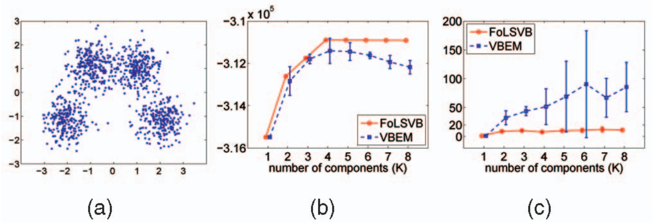

Fig. 5. Results on the synthetic data set of 1,000 500D binary data points, drawn from the mixture of four Bernoulli distributions. (a) Data set along with two dominant principle axes. (b) Lower bounds, $\widetilde{\mathcal{F}}_{Q_X}$. (c) Number of iterations to meet the convergence criterion, $\frac{1}{NK}\sum_{i=1}^N \sum_{k=1}^K |\gamma_{ik}^{(t)} - \gamma_{ik}^{(t-1)}| < 10^{-9}$. The results are averaged and annotated by one standard deviation over 30 trials with different random initial $\{\gamma_{ik}\}$.

where $\tilde{\alpha}_k^{\neg i} = \alpha_k^\circ + \langle n_k^{\neg i}\rangle$, $\tilde{\beta}_{kj,(1)}^{\neg i} = \beta_{kj,(1)}^\circ + \langle \rho_{kj}^{\neg i}\rangle$, and $\tilde{\beta}_{kj,(2)}^{\neg i} = \beta_{kj,(2)}^\circ + \langle n_k^{\neg i}\rangle - \langle \rho_{kj}^{\neg i}\rangle$ with $\langle n_k^{\neg i}\rangle = \sum_{i'=1,\neq i}^N \gamma_{i'k}$ and $\langle \rho_{kj}^{\neg i}\rangle = \sum_{i'=1,\neq i}^N \gamma_{i'k} y_{i'j}$. After converging, the log model evidence is estimated by $\widetilde{\mathcal{F}}_{Q_X} = -\log h(\varphi^\circ) + \log h(\tilde{\varphi}) - \sum_{i=1}^N \sum_{k=1}^K \gamma_{ik} \log \gamma_{ik}$ with the posterior hyperparameters $\tilde{\varphi}$ incorporating all samples.

5.2 Numerical Results

In order to see different behaviors of the algorithms, we prepared a synthetic data set of 1,000 data points, each of which has 500 dimensions. This data set was generated from the mixture of four Bernoulli distributions having equal mixing coefficients $\boldsymbol{\pi} = (0.25, 0.25, 0.25, 0.25)$ and the component's mean parameters $\boldsymbol{\mu}_k = (\mu_{kj})_{j=1}^{500}$, each of which is randomly drawn from $\mu_{kj} \in \{0.3, 0.7\}$. Especially, $\boldsymbol{\mu}_k$ was restricted to having 50 distinct elements from $\boldsymbol{\mu}_{k-1}$. Fig. 5a visualizes the data set onto 2D space by using the PCA dimensionality reduction technique, in which the data are well clustered by four clusters.

We set all prior hyperparameters to one such as $\alpha_k^\circ = 1$, $\beta_{kj,(1)}^\circ = 1$, and $\beta_{kj,(2)}^\circ = 1$. Fig. 5 shows the results based on 30 trials with different random initial $\{\gamma_{ik}\}$. Both FoLSVB and VBEM find a correct model with $K = 4$. However, in all cases, VBEM not only gives a looser lower bound than FoLSVB (Fig. 5b) but also requires more iterations to converge (Fig. 5c). In contrast to VBEM, FoLSVB shows a very low sensitivity to the initial conditions, showing an almost zero variance over 30 trials. Table 3 shows the component's responsibilities to explain the data set on the model with $K = 8$. Especially for this data set, FoLSVB uses only four mixture components to explain the four data clusters, but VBEM tends to fit all mixture components to the data points.

6 Conclusion

In this paper, we have introduced an LSVB approximate inference framework for the conjugate-exponential family. LSVB does not

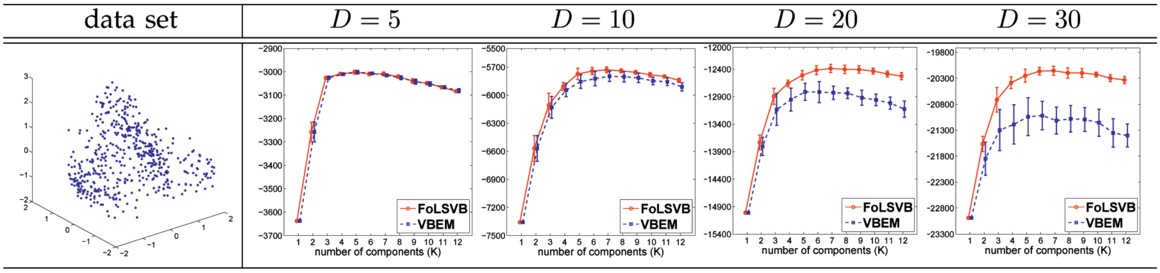

Fig. 4. Demonstration of the sensitivity to data dimension, based on the numeral digit "1" data set of 500 data points, by reducing the original 64-dimensions to be 5, 10, 20, and 30 by the PCA dimensionality reduction technique. The first figure shows the data set along with three dominant principle axes. The next four figures show the lower bounds, $\widetilde{\mathcal{F}}_{Q_X}$, as the dimension of data increases, where the results are averaged and annotated by one standard deviation over 30 trials with different initial $\{\gamma_{ik}\}$ by k-means algorithm.

TABLE 3
Component's Responsibility

	$\langle n_k \rangle = \sum_{i=1}^{N} \gamma_{ik}$ over $k = 1, 2, \ldots, 8$			
FoLSVB	269.8915	248.3201	247.4221	234.3663
	0.0000	0.0000	0.0000	0.0000
VBEM	249.2093	242.7431	234.9338	215.2370
	31.8830	14.9960	9.9977	1.0000

involve overfitting the posterior hyperparameters, a problem in VBEM, as it integrates out the model parameters. We showed that the LSVB approach gives better estimates of the model evidence as well as the distribution over the latent variables than the VBEM approach.

However, the exact implementation of LSVB is hard to do in general and we therefore presented the practical FoLSVB approximation. Through numerical results on the MoG and the MoB, we confirmed the useful behaviors of the proposed FoLSVB over the standard VBEM with the same computational cost such as faster convergence, lower sensitivity to initial conditions, and better performance with high-dimensional data. We never saw a nonconverging case with the FoLSVB algorithm in all of our examples. We conclude that our method will also be promising for other latent variable models in the conjugate-exponential family.

Appendix A
Proof of Theorem 1

First, setting the functional derivative of $\mathcal{F}_{Q_X,\tilde{\boldsymbol{\nu}}}$ with respect to $Q_{\boldsymbol{x}_i}$ given $\tilde{\boldsymbol{\nu}}$ to zero gives the optimal $Q_{\boldsymbol{x}_i}(\boldsymbol{x}_i)$ maximizing $\mathcal{F}_{Q_X,\tilde{\boldsymbol{\nu}}}$ in the form of $Q_{\boldsymbol{x}_i}(\boldsymbol{x}_i) \propto f(\boldsymbol{y}_i, \boldsymbol{x}_i) \exp\{\mathbf{m}_{\boldsymbol{\phi}}(\tilde{\boldsymbol{\nu}})^{\top} \mathbf{u}_i(\boldsymbol{y}_i, \boldsymbol{x}_i)\}$. From the definition $\mathbf{m}_{\phi}(\tilde{\boldsymbol{\nu}}) \equiv \langle \boldsymbol{\phi}(\boldsymbol{\theta}) \rangle_{P(\boldsymbol{\theta}|\eta,\tilde{\boldsymbol{\nu}})}$, this reduces to the solution in the VBE step given in [14]. Next, from the property of the convex function $\log h$, we have

$$\log h\Big(\eta, \langle \boldsymbol{\nu} \rangle_{Q_X}\Big) = \log \tilde{h}\Big(\eta, \langle \boldsymbol{\nu} \rangle_{Q_X}; \langle \boldsymbol{\nu} \rangle_{Q_X}\Big) \geq \log \tilde{h}\Big(\eta, \langle \boldsymbol{\nu} \rangle_{Q_X}; \tilde{\boldsymbol{\nu}}\Big).$$

Since $\tilde{\boldsymbol{\nu}}$ is only associated with the term $\log \tilde{h}$, it is obvious from the inequality above that, given Q_X, the optimal $\tilde{\boldsymbol{\nu}}$ maximizing $\mathcal{F}_{Q_X,\tilde{\boldsymbol{\nu}}}$ is given by $\langle \boldsymbol{\nu} \rangle_{Q_X}$, which also reduces the solution in the VBM step given in [14]. □

Appendix B
Standard Distributions

- Dirichlet distribution:

$$\mathrm{D}\Big(\boldsymbol{\pi} = (\pi_k)_{k=1}^{K} | \boldsymbol{\alpha} = (\alpha_k)_{k=1}^{K}\Big) = \frac{\Gamma\Big(\sum_{k=1}^{K} \alpha_k\Big)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} {\pi_k}^{\alpha_k - 1}.$$

- Gaussian distribution:

$$\mathrm{N}(\boldsymbol{y}|\boldsymbol{\mu}, \boldsymbol{\lambda}) = (2\pi)^{-\frac{D}{2}} |\boldsymbol{\lambda}|^{\frac{1}{2}} \exp\bigg\{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\lambda} (\boldsymbol{y}-\boldsymbol{\mu})\bigg\}.$$

- Wishart distribution:

$$\mathrm{W}(\boldsymbol{\lambda}|r, \boldsymbol{\beta}) = \frac{\pi^{-D(D-1)/4} |\boldsymbol{\beta}|^{r}}{\prod_{l=1}^{D} \Gamma\big(r + \frac{1-l}{2}\big)} |\boldsymbol{\lambda}|^{r - \frac{(D+1)}{2}} \exp\big\{-\mathrm{tr}[\boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{\lambda}]\big\}.$$

- Student distribution:

$$\mathbf{S}(\boldsymbol{y}|\boldsymbol{\mu}, \boldsymbol{\lambda}, r) = \frac{\Gamma\big(\frac{r+D}{2}\big) |\boldsymbol{\lambda}|^{1/2}}{\Gamma\big(\frac{r}{2}\big)(r\pi)^{D/2}} \bigg(1 + \frac{1}{r}(\boldsymbol{y}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\lambda} (\boldsymbol{y}-\boldsymbol{\mu})\bigg)^{-\frac{(r+D)}{2}}.$$

- Beta distribution:

$$\mathrm{Beta}\big(\mu|\beta_{(1)}, \beta_{(2)}\big) = \frac{\Gamma\big(\beta_{(1)} + \beta_{(2)}\big)}{\Gamma\big(\beta_{(1)}\big)\Gamma\big(\beta_{(2)}\big)} \mu^{\beta_{(1)}-1} (1-\mu)^{\beta_{(2)}-1}.$$

References

[1] W.H. Jefferys and J.O. Berger, "Occam's Razor and Bayesian Analysis," *Am. Scientist,* vol. 80, pp. 64-72, 1992.

[2] C.E. Rasmussen and Z. Ghahramani, "Occam's Razor," *Advances in Neural Information Processing Systems,* vol. 13, MIT Press, 2001.

[3] D.M. Chickering and D. Heckerman, "Efficient Approximation for the Marginal Likelihood of Bayesian Networks with Hidden Variables," *Machine Learning,* vol. 29, no. 2, pp. 181-212, 1997.

[4] J.M. Bernardo and A.F.M. Smith, *Bayesian Theory.* John Wiley & Sons, 2000.

[5] A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin, *Bayesian Data Analysis.* Chapman and Hall/CRC, 1995.

[6] C.M. Bishop, *Pattern Recognition and Machine Learning.* Springer, 2006.

[7] R.M. Neal, "Probabilistic Inference Using Markov Chain Monte Carlo Methods," Technical Report CRG-TR-93-1, Dept. of Computer Science, Univ. of Toronto, 1993.

[8] C.P. Robert and G. Casella, *Monte Carlo Statistical Methods.* Springer-Verlag, 1999.

[9] C. Andrieu, N.D. Freitas, A. Doucet, and M.I. Jordan, "An Introduction to MCMC for Machine Learning," *Machine Learning,* vol. 50, pp. 5-43, 2003.

[10] D.J. Mackay, ed., *Information Theory, Inference, and Learning Algorithms.* Cambridge Univ. Press, 2003.

[11] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul, "An Introduction to Variational Methods for Graphical Models," *Machine Learning,* vol. 72, no. 2, pp. 183-233, 1999.

[12] T. Jaakkola, "Tutorial on Variational Approximation Methods," *Advanced Mean Field Methods: Theory and Practice,* M. Opper and D. Saad, eds., MIT Press, 2000.

[13] H. Attias, "Variational Bayesian Framework for Graphical Models," *Advances in Neural Information Processing Systems,* vol. 12, MIT Press, 2000.

[14] Z. Ghahramani and M.J. Beal, "Propagation Algorithms for Variational Bayesian Learning," *Advances in Neural Information Processing Systems,* vol. 13, MIT Press, 2001.

[15] Z. Ghahramani and M.J. Beal, "Variational Inference for Bayesian Mixtures of Factor Analysers," *Advances in Neural Information Processing Systems,* vol. 12, MIT Press, 2000.

[16] M.J. Beal and Z. Ghahramani, "The Variational Bayesian EM Algorithm for Incomplete Data: With Application to Scoring Graphical Model Structures," *Bayesian Statistics,* vol. 7, Oxford Univ. Press, 2003.

[17] N. Ueda and Z. Ghahramani, "Bayesian Model Search for Mixture Models Based on Optimizing Variational Bounds," *Neural Networks,* vol. 15, pp. 1223-1241, 2002.

[18] J. Winn and C. Bishop, "Variational Message Passing," *J. Machine Learning Research,* vol. 6, pp. 661-694, 2005.

[19] M.J. Beal, "Variational Algorithms for Approximate Bayesian Inference," PhD dissertation, Gatsby Computational Neuroscience Unit, Univ. College London, 2003.

[20] Y.W. Teh, D. Newman, and M. Welling, "A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation," *Advances in Neural Information Processing Systems,* vol. 19, 2007.

[21] Y.W. Teh, K. Kurihara, and M. Welling, "Collapsed Variational Inference for HDP," *Advances in Neural Information Processing Systems,* vol. 20, 2008.

[22] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc. B,* vol. 39, pp. 1-38, 1977.

[23] G. McLachlan and D. Peel, *Finite Mixture Models.* Wiley-Interscience, 2000.

[24] S. Richardson and P.J. Green, "On Bayesian Analysis of Mixtures with an Unknown Number of Components," *J. Royal Statistical Soc. B,* vol. 59, pp. 731-792, 1997.

[25] A. Asuncion and D. Newman, *UCI Machine Learning Repository,* http://www.ics.uci.edu/~mlearn/MLRepository.html, 2007.

[26] M. Svensén and C.M. Bishop, "Robust Bayesian Mixture Modelling," *Neurocomputing,* vol. 64, pp. 235-252, 2004.