Journal of Machine Learning Research 11 (2010) 3269-3311 Submitted 11/08; Revised 6/10; Published 12/10

Classification with Incomplete Data Using Dirichlet Process Priors

Chunping Wang CW36@EE.DUKE.EDU
Xuejun Liao XJLIAO@EE.DUKE.EDU
Lawrence Carin LCARIN@EE.DUKE.EDU
Department of Electrical and Computer Engineering
Duke University
Durham, NC 27708-0291, USA

David B. Dunson DUNSON@STAT.DUKE.EDU
Department of Statistical Science
Duke University
Durham, NC 27708-0291, USA

Editor: David Blei

Abstract

A non-parametric hierarchical Bayesian framework is developed for designing a classifier, based on a mixture of simple (linear) classifiers. Each simple classifier is termed a local "expert", and the number of experts and their construction are manifested via a Dirichlet process formulation. The simple form of the "experts" allows analytical handling of incomplete data. The model is extended to allow simultaneous design of classifiers on multiple data sets, termed multi-task learning, with this also performed non-parametrically via the Dirichlet process. Fast inference is performed using variational Bayesian (VB) analysis, and example results are presented for several data sets. We also perform inference via Gibbs sampling, to which we compare the VB results.

Keywords: classification, incomplete data, expert, Dirichlet process, variational Bayesian, multi-task learning

1. Introduction

In many applications one must deal with data that have been collected incompletely. For example, in censuses and surveys, some participants may not respond to certain questions (Rubin, 1987); in email spam filtering, server information may be unavailable for emails from external sources (Dick et al., 2008); in medical studies, measurements on some subjects may be partially lost at certain stages of the treatment (Ibrahim, 1990); in DNA analysis, gene-expression microarrays may be incomplete due to insufficient resolution, image corruption, or simply dust or scratches on the slide (Wang et al., 2006); in sensing applications, a subset of sensors may be absent or fail to operate at certain regions (Williams and Carin, 2005). Unlike in semi-supervised learning (Ando and Zhang, 2005) where missing labels (responses) must be addressed, features (inputs) are partially missing in the aforementioned incomplete-data problems. Since most data analysis procedures (for example, regression and classification) are designed for complete data, and cannot be directly applied to incomplete data, the appropriate handling of missing data is challenging.

Traditionally, data are often "completed" by *ad hoc* editing, such as case deletion and single imputation, where feature vectors with missing values are simply discarded or completed with specific values in the initial stage of analysis, before the main inference (for example, mean imputation and regression imputation see Schafer and Graham, 2002). Although analysis procedures designed for complete data become applicable after these edits, shortcomings are clear. For case deletion, discarding information is generally inefficient, especially when data are scarce. Secondly, the remaining complete data may be statistically unrepresentative. More importantly, even if the incomplete-data problem is eliminated by ignoring data with missing features in the training phase, it is still inevitable in the test stage since test data cannot be ignored simply because a portion of features are missing. For single imputation, the main concern is that the uncertainty of the missing features is ignored by imputing fixed values.

The work of Rubin (1976) developed a theoretical framework for incomplete-data problems, where widely-cited terminology for missing patterns was first defined. It was proven that ignoring the *missing mechanism* is appropriate (Rubin, 1976) under the *missing at random* (MAR) assumption, meaning that the *missing mechanism* is conditionally independent of the missing features given the observed data. As elaborated later, given the MAR assumption (Dick et al., 2008; Ibrahim, 1990; Williams and Carin, 2005), incomplete data can generally be handled by full maximum likelihood and Bayesian approaches; however, when the *missing mechanism* does depend on the missing values (*missing not at random* or MNAR), a problem-specific model is necessary to describe the *missing mechanism*, and no general approach exists. In this paper, we address missing features under the MAR assumption. Previous work in this setting may be placed into two groups, depending on whether the missing data are handled before algorithm learning or within the algorithm.

For the former, an extra step is required to estimate $p(\boldsymbol{x}^m|\boldsymbol{x}^o)$, conditional distributions of missing values given observed ones, with this step distinct from the main inference algorithm. After $p(\boldsymbol{x}^m|\boldsymbol{x}^o)$ is learned, various imputation methods may be performed. As a Monte Carlo approach, Bayesian multiple imputation (MI) (Rubin, 1987) is widely used, where multiple ($M > 1$) samples from $p(\boldsymbol{x}^m|\boldsymbol{x}^o)$ are imputed to form M "complete" data sets, with the complete-data algorithm applied on each, and results of those imputed data sets combined to yield a final result. The MI method "completes" data sets so that algorithms designed for complete data become applicable. Furthermore, Rubin (1987) showed that MI does not require as many samples as Monte Carlo methods usually do. With a mild Gaussian mixture model (GMM) assumption for the joint distribution of observed and missing data, Williams et al. (2007) managed to analytically integrate out missing values over $p(\boldsymbol{x}^m|\boldsymbol{x}^o)$ and performed essentially infinite imputations. Since explicit imputations are avoided, this method is more efficient than the MI method, as suggested by empirical results (Williams et al., 2007). Other examples of these two-step methods include Williams and Carin (2005), Smola et al. (2005) and Shivaswamy et al. (2006).

The other class of methods explicitly addresses missing values during the model-learning procedure. The work proposed by Chechik et al. (2008) represents a special case, in which no model is assumed for *structurally absent* values; the margin for the support vector machine (SVM) is re-scaled according to the observed features for each instance. Empirical results (Chechik et al., 2008) show that this procedure is comparable to several single-imputation methods when values are *missing at random*. Another recent work (Dick et al., 2008) handles the missing features inside the procedure of learning a support vector machine (SVM), without constraining the distribution of missing features to any specific class. The main concern is that this method can only handle missing features in the training data; however, in many applications one cannot control whether missing values occur in the training or test data.

A widely employed approach for handling missing values within the algorithm involves maximum likelihood (ML) estimation via expectation maximization (EM) (Dempster et al., 1977). Besides the latent variables (e.g., mixture component indicators), the missing features are also integrated out in the E-step so that the likelihood is maximized with respect to model parameters in the M-step. The main difficulty is that the integral in the E-step is analytically tractable only when an assumption is made on the distribution of the missing features. For example, the intractable integral is avoided by requiring the features to be discrete (Ibrahim, 1990), or assuming a Gaussian mixture model (GMM) for the features (Ghahramani and Jordan, 1994; Liao et al., 2007). The discreteness requirement is often too restrictive, while the GMM assumption is mild since it is well known that a GMM can approximate arbitrary continuous distributions.

In Liao et al. (2007) the authors proposed a quadratically gated mixture of experts (QGME) where the GMM is used to form the gating network, statistically partitioning the feature space into quadratic subregions. In each subregion, one linear classifier works as a local “expert”. As a mixture of experts (Jacobs et al., 1991), the QGME is capable of addressing a classification problem with a nonlinear decision boundary in terms of multiple local experts; the simple form of this model makes it straightforward to handle incomplete data without completing kernel functions (Graepel, 2002; Williams and Carin, 2005). However, as in many mixture-of-expert models (Jacobs et al., 1991; Waterhouse and Robinson, 1994; Xu et al., 1995), the number of local experts in the QGME must be specified initially, and thus a model-selection stage is in general necessary. Moreover, since the expectation-maximization method renders a point (single) solution that maximizes the likelihood, over-fitting may occur when data are scarce relative to the model complexity.

In this paper, we first extend the finite QGME (Liao et al., 2007) to an infinite QGME (iQGME), with theoretically an infinite number of experts realized via a Dirichlet process (DP) (Ferguson, 1973) prior; this yields a fully Bayesian solution, rather than a point estimate. In this manner model selection is avoided and the uncertainty on the number of experts is captured in the posterior density function.

The Dirichlet process (Ferguson, 1973) has been an active topic in many applications since the middle 1990s, for example, density estimation (Escobar and West, 1995; MacEachern and Müller, 1998; Dunson et al., 2007) and regression/curve fitting (Müller et al., 1996; Rasmussen and Ghahramani, 2002; Meeds and Osindero, 2006; Shahbaba and Neal, 2009; Rodríguez et al., 2009; Hannah et al., 2010). The latter group is relevant to classification problems of interest in this paper. The work in Müller et al. (1996) jointly modeled inputs and responses as a Dirichlet process mixture of multivariate normals, while Rodríguez et al. (2009) extended this model to simultaneously estimate multiple curves using dependent DP. In Rasmussen and Ghahramani (2002) and Meeds and Osindero (2006) two approaches to constructing infinite mixtures of Gaussian Process (GP) experts were proposed. The difference is that Meeds and Osindero (2006) specified the gating network using a multivariate Gaussian mixture instead of a (fixed) input-dependent Dirichlet Process. In Shahbaba and Neal (2009) another form of infinite mixtures of experts was proposed, where experts are specified by a multinomial logit (MNL) model (also called softmax) and the gating network is Gaussian mixture model with independent covariates. Further, Hannah et al. (2010) generalized existing DP-based nonparametric regression models to accommodate different types of covariates and responses, and further gave theoretical guarantees for this class of models.

Our focus in this paper is on developing classification models that handle incomplete inputs/covariates efficiently using Dirichlet process. Some of the above Dirichlet process regression models are potentially capable of handling incomplete inputs/features; however, none of them actually deal with such problems. In Müller et al. (1996), although the joint multivariate normal assumption over inputs and responses endow this approach with the potential of handling missing features and/or missing responses naturally, a good estimation for the joint distribution does not guarantee a good estimation for classification boundaries. Other than a full joint Gaussian distribution assumption, explicit classifiers were used to model the conditional distribution of responses given covariates in the models proposed in Meeds and Osindero (2006) and Shahbaba and Neal (2009). These two models are highly related to the iQGME proposed here. The independence assumption of covariates in Shahbaba and Neal (2009) leads to efficient computation but is not appealing for handling missing features. With Gaussian process experts (Meeds and Osindero, 2006), the inference for missing features is not analytical for fast inference algorithms such as variational Bayesian (Beal, 2003) and EM, and the computation could be prohibitive for large data sets. The iQGME seeks a balance between the ease of inference, computational burden and the ability of handling missing features. For high-dimensional data sets, we develop a variant of our model based on mixtures of factor analyzers (MFA) (Ghahramani and Hinton, 1996; Ghahramani and Beal, 2000), where a low-rank assumption is made for the covariance matrices of high-dimensional inputs in each cluster.

In addition to challenges with incomplete data, one must often address an insufficient quantity of labeled data. In Williams et al. (2007) the authors employed semi-supervised learning (Zhu, 2005) to address this challenge, using the contextual information in the unlabeled data to augment the limited labeled data, all done in the presence of missing/incomplete data. Another form of context one may employ to address limited labeled data is multi-task learning (MTL) (Caruana, 1997; Ando and Zhang, 2005), which allows the learning of multiple tasks simultaneously to improve generalization performance. The work of Caruana (1997) provided an overview of MTL and demonstrated it on multiple problems. In recent research, a hierarchical statistical structure has been favored for such models, where information is transferred via a common prior within a hierarchical Bayesian model (Yu et al., 2003; Zhang et al., 2006). Specifically, information may be transferred among related tasks (Xue et al., 2007) when the Dirichlet process (DP) (Ferguson, 1973) is introduced as a common prior. To the best of our knowledge, there is no previous example of addressing incomplete data in a multi-task setting, this problem constituting an important aspect of this paper.

The main contributions of this paper may be summarized as follows. The problem of missing data in classifier design is addressed by extending QGME (Liao et al., 2007) to a fully Bayesian setting, with the number of local experts inferred automatically via a DP prior. The algorithm is further extended to a multi-task setting, again using a non-parametric Bayesian model, simultaneously learning J missing-data classification problems, with appropriate sharing (could be global or local). Throughout, efficient inference is implemented via the variational Bayesian (VB) method (Beal, 2003). To quantify the accuracy of the VB results, we also perform comparative studies based on Gibbs sampling.

The remainder of the paper is organized as follows. In Section 2 we extend the finite QGME (Liao et al., 2007) to an infinite QGME via a Dirichlet process prior. The incomplete-data problem is defined and discussed in Section 3. Extension to the multi-task learning case is considered in Section 4, and variational Bayesian inference is developed in Section 5. Experimental results for synthetic data and multiple real data sets are presented in Section 6, followed in Section 7 by conclusions and a discussions of future research directions.

2. Infinite Quadratically Gated Mixture of Experts

In this section, we first provide a brief review of the quadratically gated mixture of experts (QGME) (Liao et al., 2007) and Dirichlet process (DP) (Ferguson, 1973), and then extend the number of experts to be infinite via DP.

2.1 Quadratically Gated Mixture of Experts

Consider a binary classification problem with real-valued P-dimensional column feature vectors $\boldsymbol{x}_i$ and corresponding class labels $y_i \in \{1, -1\}$. We assume binary labels for simplicity, while the proposed method may be directly extended to cases with more than two classes. Latent variables t_i are introduced as “soft labels” associated with y_i, as in probit models (Albert and Chib, 1993), where $y_i = 1$ if $t_i > 0$ and $y_i = -1$ if $t_i \leq 0$. The finite quadratically gated mixture of experts (QGME) (Liao et al., 2007) is defined as

$$\begin{aligned}(t_i|z_i = h) &\sim \mathcal{N}(\boldsymbol{w}_h^T \boldsymbol{x}_i^b, 1), & (1)\\ (\boldsymbol{x}_i|z_i = h) &\sim \mathcal{N}_P(\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h^{-1}), & (2)\\ (z_i|\boldsymbol{\pi}) &\sim \sum_{h=1}^{K} \pi_h \delta_h, & (3)\end{aligned}$$

with $\sum_{h=1}^{K} \pi_h = 1$, and where δ_h is a point measure concentrated at h (with probability one, a draw from δ_h will be h). The $(P+1) \times K$ matrix $\boldsymbol{W}$ has columns $\boldsymbol{w}_h$, where each $\boldsymbol{w}_h$ are the weights on a local linear classifier, and the $\boldsymbol{x}_i^b$ are feature vectors with an intercept, that is, $\boldsymbol{x}_i^b = [\boldsymbol{x}_i^T, 1]^T$. A total of K groups of $\boldsymbol{w}_h$ are introduced to parameterize the K experts. With probability π_h the indicator for the ith data point satisfies $z_i = h$, which means the hth local expert is selected, and $\boldsymbol{x}_i$ is distributed according to a P-variate Gaussian distribution with mean $\boldsymbol{\mu}_h$ and precision $\boldsymbol{\Lambda}_h$.

It can be seen that the QGME is highly related to the mixture of experts (ME) (Jacobs et al., 1991) and the hierarchical mixture of experts (HME) (Jordan and Jacobs, 1994) if we write the conditional distribution of labels as

$$p(y_i|\boldsymbol{x}_i) = \sum_{h=1}^{K} p(z_i = h|\boldsymbol{x}_i) p(y_i|z_i = h, \boldsymbol{x}_i), \tag{4}$$

where

$$\begin{aligned}p(y_i|z_i = h, \boldsymbol{x}_i) &= \int_{t_i y_i > 0} \mathcal{N}(t_i|\boldsymbol{w}_h^T \boldsymbol{x}_i^b, 1) dt_i, & (5)\\ p(z_i = h|\boldsymbol{x}_i) &= \frac{\pi_h \mathcal{N}_P(\boldsymbol{x}_i|\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h^{-1})}{\sum_{k=1}^{K} \pi_k \mathcal{N}_P(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})}. & (6)\end{aligned}$$

From (4), as a special case of the ME, the QGME is capable of handling nonlinear problems with linear experts characterized in (5). However, unlike other ME models, the QGME probabilistically partitions the feature space through a mixture of K Gaussian distributions for $\boldsymbol{x}_i$ as in (6). This assumption on the distribution of $\boldsymbol{x}_i$ is mild since it is well known that a Gaussian mixture model (GMM) is general enough to approximate any continuous distribution. In the QGME, $\boldsymbol{x}_i$ as well as y_i are treated as random variables (generative model) and we consider a joint probability $p(y_i, \boldsymbol{x}_i)$ instead of a conditional probability $p(y_i|\boldsymbol{x}_i)$ for fixed $\boldsymbol{x}_i$ as in most ME models (which are typically discriminative). Previous work on the comparison between discriminative and generative models may be found in Ng and Jordan (2002) and Liang and Jordan (2008). In the QGME, the GMM of the inputs $\boldsymbol{x}_i$ plays two important roles: (*i*) as a gating network, while (*ii*) enabling analytic incorporation of incomplete data during classifier inference (as discussed further below).

The QGME (Liao et al., 2007) is inferred via the expectation-maximization (EM) method, which renders a point-estimate solution for an initially specified model (1)-(3), with a fixed number K of local experts. Since learning the correct model requires model selection, and moreover in many applications there may exist no such fixed "correct" model, in the work reported here we infer the full posterior for a QGME model with the number of experts data-driven. The objective can be achieved by imposing a nonparametric Dirichlet process (DP) prior.

2.2 Dirichlet Process

The Dirichlet process (DP) (Ferguson, 1973) is a random measure defined on measures of random variables, denoted as $\mathcal{DP}(\alpha G_0)$, with a real scaling parameter $\alpha \geq 0$ and a base measure G_0. Assuming that a measure is drawn $G \sim \mathcal{DP}(\alpha G_0)$, the base measure G_0 reflects the prior expectation of G and the scaling parameter α controls how much G is allowed to deviate from G_0. In the limit $\alpha \to \infty$, G goes to G_0; in the limit $\alpha \to 0$, G reduces to a delta function at a random point in the support of G_0.

The stick-breaking construction (Sethuraman, 1994) provides an explicit form of a draw from a DP prior. Specifically, it has been proven that a draw G may be constructed as

$$G = \sum_{h=1}^{\infty} \pi_h \delta_{\boldsymbol{\theta}_h^*}, \tag{7}$$

with $0 \leq \pi_h \leq 1$ and $\sum_{h=1}^{\infty} \pi_h = 1$, and

$$\pi_h = V_h \prod_{l=1}^{h-1}(1-V_l), \quad V_h \overset{iid}{\sim} Be(1,\alpha), \quad \boldsymbol{\theta}_h^* \overset{iid}{\sim} G_0.$$

From (7), it is clear that G is discrete (with probability one) with an infinite set of weights π_h at atoms $\boldsymbol{\theta}_h^*$. Since the weights π_h decrease stochastically with h, the summation in (7) may be truncated with N terms, yielding an N-level truncated approximation to a draw from the Dirichlet process (Ishwaran and James, 2001).

Assuming that underlying variables $\boldsymbol{\theta}_i$ are drawn i.i.d. from G, the associated data $\chi_i \sim F(\boldsymbol{\theta}_i)$ will naturally cluster with $\boldsymbol{\theta}_i$ taking distinct values $\boldsymbol{\theta}_h^*$, where the function $F(\boldsymbol{\theta})$ represents an arbitrary parametric model for the observed data, with hidden parameters $\boldsymbol{\theta}$. Therefore, the number of clusters is automatically determined by the data and could be "infinite" in principle. Since $\boldsymbol{\theta}_i$ take distinct values $\boldsymbol{\theta}_h^*$ with probabilities π_h, this clustering is a statistical procedure instead of a hard partition, and thus we only have a belief on the number of clusters, which is affected by the scaling parameter α. As the value of α influences the prior belief on the clustering, a gamma hyper-prior is usually employed on α.

2.3 Infinite QGME via DP

Consider a classification task with a training data set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) : i = 1, \ldots, n\}$, where $\boldsymbol{x}_i \in \mathbb{R}^P$ and $y_i \in \{-1, 1\}$. With soft labels t_i introduced as in Section 2.1, the infinite QGME (iQGME) model is achieved via a DP prior imposed on the measure G of $(\boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i, \boldsymbol{w}_i)$, the hidden variables characterizing the density function of each data point $(\boldsymbol{x}_i, t_i)$. For simplicity, the same symbols are used to denote parameters associated with each data point and the distinct values, with subscripts i and h indexing data points and unique values, respectively:

$$\begin{aligned}
(\boldsymbol{x}_i, t_i) &\sim \mathcal{N}_P(\boldsymbol{x}_i|\boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i^{-1})\mathcal{N}(t_i|\boldsymbol{w}_i^T\boldsymbol{x}_i^b, 1), \\
(\boldsymbol{\mu}_i, \boldsymbol{\Lambda}_i, \boldsymbol{w}_i) &\overset{iid}{\sim} G, \\
G &\sim \mathcal{DP}(\alpha G_0), \qquad (8)
\end{aligned}$$

where the base measure G_0 is factorized as the product of a normal-Wishart prior for $(\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h)$ and a normal prior for $\boldsymbol{w}_h$, for the sake of conjugacy. As discussed in Section 2.2, data samples cluster automatically, and the same mean $\boldsymbol{\mu}_h$, covariance matrix $\boldsymbol{\Lambda}_h$ and regression coefficients (expert) $\boldsymbol{w}_h$ are shared for a given cluster h. Using the stick-breaking construction, we elaborate (8) as follows for $i = 1, \ldots, n$ and $h = 1, \ldots, \infty$:

Data generation:

$$\begin{aligned}
(t_i|z_i = h) &\sim \mathcal{N}(\boldsymbol{w}_h^T\boldsymbol{x}_i^b, 1), \\
(\boldsymbol{x}_i|z_i = h) &\sim \mathcal{N}_P(\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h^{-1}),
\end{aligned}$$

Drawing indicators:

$$\begin{aligned}
z_i &\sim \sum_{h=1}^{\infty} \pi_h \delta_h, \quad \text{where} \quad \pi_h = V_h \prod_{l<h}(1 - V_l), \\
V_h &\sim Be(1, \alpha),
\end{aligned}$$

Drawing parameters from G_0 :

$$\begin{aligned}
(\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h) &\sim \mathcal{N}_P(\boldsymbol{\mu}_h|\boldsymbol{m}_0, u_0^{-1}\boldsymbol{\Lambda}_h^{-1})\mathcal{W}(\boldsymbol{\Lambda}_h|\boldsymbol{B}_0, \nu_0), \\
\boldsymbol{w}_h &\sim \mathcal{N}_{P+1}(\boldsymbol{\zeta}, [\text{diag}(\boldsymbol{\lambda})]^{-1}), \quad \text{where} \quad \boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_{P+1}].
\end{aligned}$$

Furthermore, to achieve a more robust algorithm, we assign diffuse hyper-priors on several crucial parameters. As discussed in Section 2.2, the scaling parameter α reflects our prior belief on the number of clusters. For the sake of conjugacy, a diffuse Gamma prior is usually assumed for α as suggested by West et al. (1994). In addition, parameters $\boldsymbol{\zeta}, \boldsymbol{\lambda}$ characterizing the prior of the distinct local classifiers $\boldsymbol{w}_h$ are another set of important parameters, since we focus on classification tasks. Normal-Gamma priors are the conjugate priors for the mean and precision of a normal density. Therefore,

$$\begin{aligned}
\alpha &\sim Ga(\tau_{10}, \tau_{20}), \\
(\boldsymbol{\zeta}|\boldsymbol{\lambda}) &\sim \mathcal{N}_{P+1}(\boldsymbol{0}, \gamma_0^{-1}[\text{diag}(\boldsymbol{\lambda})]^{-1}), \\
\lambda_p &\sim Ga(a_0, b_0), \quad p = 1, \ldots, P+1,
\end{aligned}$$

where $\tau_{10}, \tau_{20}, a_0, b_0$ are usually set to be much less than one and of about the same magnitude, so that the constructed Gamma distributions with means about one and large variances are diffuse; γ_0 is usually set to be around one.

The graphical representation of the iQGME for single-task learning is shown in Figure 1. We notice that a possible variant with sparse local classifiers could be obtained if we impose zero mean for the local classifiers $\boldsymbol{w}_h$, that is, $\boldsymbol{\zeta} = \boldsymbol{0}$, and retain the Gamma hyper-prior for the precision $\boldsymbol{\lambda}$, as

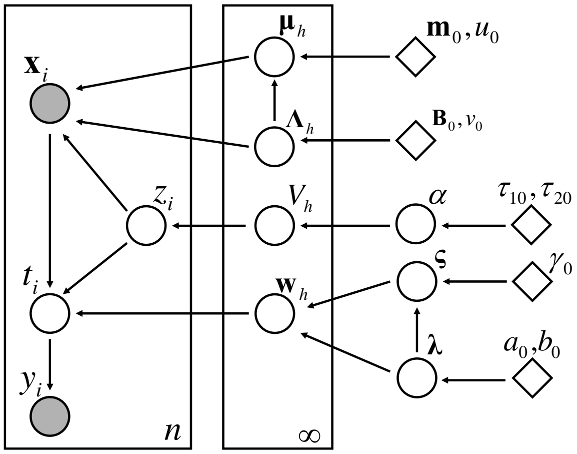


Figure 1: Graphical representation of the iQGME for single-task leaning. All circles denote random variables, with shaded ones indicating observable data, and bright ones representing hidden variables. Diamonds denote fixed hyper-parameters, boxes represent independent replicates with numbers at the lower right corner indicating the numbers of i.i.d. copies, and arrows indicate the dependence between variables (pointing from parents to children).

in the relevance vector machine (RVM) (Tipping, 2000), which employs a corresponding Student-t sparseness prior on the weights. Although this sparseness prior is useful for seeking relevant features in many applications, imposing the same sparse pattern for all the local experts is not desirable.

2.4 Variant for High-Dimensional Problems

For the classification problem, we assume access to a training data set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) : i = 1, \ldots, n\}$, where feature vectors $\boldsymbol{x}_i \in \mathbb{R}^P$ and labels $y_i \in \{-1, 1\}$. We have assumed that the feature vectors of objects in cluster h are generated from a P-variate normal distribution with mean $\boldsymbol{\mu}_h$ and covariance matrix $\boldsymbol{\Lambda}_h^{-1}$, that is,

$$(\boldsymbol{x}_i | z_i = h) \quad \sim \quad \mathcal{N}_P(\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h^{-1}) \tag{9}$$

It is well known that each covariance matrix has $P(P+1)/2$ parameters to be estimated. Without any further assumption, the estimation of these parameters could be computationally prohibitive for large P, especially when the number of available training data n is small, which is common for classification applications. By imposing an approximately low-rank constraint on the covariances, as in well-studied mixtures of factor analyzers (MFA) models (Ghahramani and Hinton, 1996; Ghahramani and Beal, 2000), the number of unknowns could be significantly reduced. Specifically, assume a vector of standard normal latent factors $\boldsymbol{s}_i \in \mathbb{R}^{T \times 1}$ for data $\boldsymbol{x}_i$, a factor loading matrix $\boldsymbol{A}_h \in \mathbb{R}^{P \times T}$ for cluster h, and Gaussian residues $\boldsymbol{\epsilon}_i$ with diagonal covariance matrix $\psi_h \boldsymbol{I}_P$, then

$$(\boldsymbol{x}_i | z_i = h) \quad \sim \quad \mathcal{N}_P(\boldsymbol{A}_h \boldsymbol{s}_i + \boldsymbol{\mu}_h, \psi_h^{-1} \boldsymbol{I}_P).$$

Marginalizing $\boldsymbol{s}_i$ with $\boldsymbol{s}_i \sim \mathcal{N}_T(\mathbf{0}, \boldsymbol{I}_T)$, we recover (9), with $\boldsymbol{\Lambda}_h^{-1} = \boldsymbol{A}_h \boldsymbol{A}_h^T + \psi_h^{-1} \boldsymbol{I}_P$. The number of free parameters is significantly reduced if $T << P$.

In this paper, we modify the MFA model for classification applications with scarce samples. First, we consider a common loading matrix $\boldsymbol{A}$ for all the clusters, and introduce a binary vector $\boldsymbol{b}_h$ for each cluster to select which columns of $\boldsymbol{A}$ are used, that is,

$$(\boldsymbol{x}_i|z_i=h) \quad \sim \quad \mathcal{N}_P(\boldsymbol{A}\text{diag}(\boldsymbol{d}\circ\boldsymbol{b}_h)\boldsymbol{s}_i+\boldsymbol{\mu}_h,\psi_h^{-1}\boldsymbol{I}_P),$$

where each column of $\boldsymbol{A}$, $\boldsymbol{A}_l \sim \mathcal{N}_P(\boldsymbol{0},P^{-1}\boldsymbol{I}_P)$, $\boldsymbol{s}_i \sim \mathcal{N}_L(\boldsymbol{0},\boldsymbol{I}_L)$, $\boldsymbol{d}$ is a vector responsible for scale, and $\circ$ is a component-wise (Hadamard) product. For $\boldsymbol{d}$ we employ the prior $d_l \sim \mathcal{N}(0,\beta_l^{-1})$ with $\beta_l \sim Ga(c_0,d_0)$. Furthermore, we let the algorithm infer the intrinsic number of factors by imposing a low-rank belief for each cluster through the prior of $\boldsymbol{b}_h$, that is,

$$b_{hl} \quad \sim \quad Bern(\pi_{hl}), \quad \pi_{hl} \sim Be(a_0/L,b_0(L-1)/L), \quad l=1,\ldots,L,$$

where L is a large number, which defines the largest possible dimensionality the algorithm may infer. Through the choice of a_0 and b_0 we impose our prior belief about the intrinsic dimensionality of cluster h (upon integrating out the draw $\boldsymbol{\pi}_h$, the number of non-zero components of $\boldsymbol{b}_h$ is drawn from $Binomial[L,a_0/(a_0+b_0(L-1))]$). As a result, both the number of clusters and the dimensionality of each cluster is inferred by this variant of iQGME.

With this form of iQGME, we could build local linear classifiers in either the original feature space or the (low-dimensional) space of latent factors $\boldsymbol{s}_i$. For the sake of computational simplicity, we choose to classify in the low-dimensional factor space.

3. Incomplete Data Problem

In the above discussion it was assumed that all components of the feature vectors were available (no missing data). In this section, we consider the situation for which feature vectors $\boldsymbol{x}_i$ are partially observed. We partition each feature vector $\boldsymbol{x}_i$ into observed and missing parts, $\boldsymbol{x}_i=[\boldsymbol{x}_i^{o_i};\boldsymbol{x}_i^{m_i}]$, where $\boldsymbol{x}_i^{o_i}=\{x_{ip}:p\in o_i\}$ denotes the subvector of observed features and $\boldsymbol{x}_i^{m_i}=\{x_{ip}:p\in m_i\}$ represents the subvector of missing features, with o_i and m_i denoting the set of indices for observed and missing features, respectively. Each $\boldsymbol{x}_i$ has its own observed set o_i and missing set m_i, which may be different for each i. Following a generic notation (Schafer and Graham, 2002), we refer to $\boldsymbol{R}$ as the missingness. For an arbitrary missing pattern, $\boldsymbol{R}$ could be defined as a missing data indicator matrix, that is,

$$R_{ip}=\begin{cases} 1, & x_{ip} \text{ observed}, \\ 0, & x_{ip} \text{ missing}. \end{cases}$$

We use $\boldsymbol{\xi}$ to denote parameters characterizing the distribution of $\boldsymbol{R}$, which is usually called the *missing mechanism*. In the classification context, the joint distribution of class labels, observed features and the missingness $\boldsymbol{R}$ may be given by integrating out the missing features $\boldsymbol{x}^m$,

$$p(y,\boldsymbol{x}^o,\boldsymbol{R}|\boldsymbol{\theta},\boldsymbol{\xi})=\int p(y,\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{R}|\boldsymbol{x},\boldsymbol{\xi})d\boldsymbol{x}^m. \tag{10}$$

To handle such a problem analytically, assumptions must be made on the distribution of $\boldsymbol{R}$. If the *missing mechanism* is conditionally independent of missing values $\boldsymbol{x}^m$ given the observed data, that is, $p(\boldsymbol{R}|\boldsymbol{x},\boldsymbol{\xi})=p(\boldsymbol{R}|\boldsymbol{x}^o,\boldsymbol{\xi})$, the missing data are defined to be *missing at random* (MAR) (Rubin, 1976). Consequently, (10) reduces to

$$p(y,\boldsymbol{x}^o,\boldsymbol{R}|\boldsymbol{\theta},\boldsymbol{\xi})=p(\boldsymbol{R}|\boldsymbol{x}^o,\boldsymbol{\xi})\int p(y,\boldsymbol{x}|\boldsymbol{\theta})d\boldsymbol{x}^m=p(\boldsymbol{R}|\boldsymbol{x}^o,\boldsymbol{\xi})p(y,\boldsymbol{x}^o|\boldsymbol{\theta}). \tag{11}$$

According to (11), the likelihood is factorizable under the assumption of MAR. As long as the prior $p(\boldsymbol{\theta},\boldsymbol{\xi}) = p(\boldsymbol{\theta})p(\boldsymbol{\xi})$ (factorizable), the posterior

$$p(\boldsymbol{\theta},\boldsymbol{\xi}|y,\boldsymbol{x}^o,\boldsymbol{R}) \propto p(y,\boldsymbol{x}^o,\boldsymbol{R}|\boldsymbol{\theta},\boldsymbol{\xi})p(\boldsymbol{\theta},\boldsymbol{\xi}) = p(\boldsymbol{R}|\boldsymbol{x}^o,\boldsymbol{\xi})p(\boldsymbol{\xi})p(y,\boldsymbol{x}^o|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

is also factorizable. For the purpose of inferring model parameters $\boldsymbol{\theta}$, no explicit specification is necessary on the distribution of the missingness. As an important special case of MAR, *missing completely at random* (MCAR) occurs if we can further assume that $p(\boldsymbol{R}|\boldsymbol{x},\boldsymbol{\xi}) = p(\boldsymbol{R}|\boldsymbol{\xi})$, which means the distribution of missingness is independent of observed values $\boldsymbol{x}^o$ as well. When the *missing mechanism* depends on missing values $\boldsymbol{x}^m$, the data are termed to be *missing not at random* (MNAR). From (10), an explicit form has to be assumed for the distribution of the missingness, and both the accuracy and the computational efficiency should be concerned.

When missingness is not totally controlled, as in most realistic applications, we cannot tell from the data alone whether the MCAR or MAR assumption is valid. Since the MCAR or MAR assumption is unlikely to be precisely satisfied in practice, inference based on these assumptions may lead to a bias. However, as demonstrated in many cases, it is believed that for realistic problems departures from MAR are usually not large enough to significantly impact the analysis (Collins et al., 2001). On the other hand, without the MAR assumption, one must explicitly specify a model for the missingness $\boldsymbol{R}$, which is a difficult task in most cases. As a result, the data are typically assumed to be either MCAR or MAR in the literature, unless significant correlations between the missing values and the distribution of the missingness are suspected.

In this work we make the MAR assumption, and thus expression (11) applies. In the iQGME framework, the joint likelihood may be further expanded as

$$p(y,\boldsymbol{x}^o|\boldsymbol{\theta}) = \int p(y,\boldsymbol{x}|\boldsymbol{\theta})d\boldsymbol{x}^m = \int_{ty>0}\int p(t|\boldsymbol{x},\boldsymbol{\theta}_2)p(\boldsymbol{x}|\boldsymbol{\theta}_1)d\boldsymbol{x}^m dt. \quad (12)$$

The solution to such a problem with incomplete data $\boldsymbol{x}^m$ is analytical since the distributions of t and $\boldsymbol{x}$ are assumed to be a Gaussian and a Gaussian mixture model, respectively. Naturally, the missing features could be regarded as hidden variables to be inferred and the graphical representation of the iQGME with incomplete data remains the same as in Figure 1, except that the node presenting features are partially observed now. As elaborated below, the important but mild assumption that the features are distributed as a GMM enables us to analytically infer the variational distributions associated with the missing values in a procedure of variational Bayesian inference.

As in many models (Williams et al., 2007), estimating the distribution of the missing values first and learning the classifier at a second step gives the flexibility of selecting the classifier for the second step. However, (12) suggests that the classifier and the data distribution are coupled, provided that partial data are missing and thus have to be integrated out. Therefore, a joint estimation of missing features and classifiers (searching in the space of $(\boldsymbol{\theta_1},\boldsymbol{\theta}_2)$) is more desirable than a two-step process (searching in the space of $\boldsymbol{\theta_1}$ for the distribution of the data, and then in the space of $\boldsymbol{\theta}_2$ for the classifier).

4. Extension to Multi-Task Learning

Assume we have J data sets, with the jth represented as $\mathcal{D}_j = \{(\boldsymbol{x}_{ji}, y_{ji}) : i = 1,\ldots,n_j\}$; our goal is to design a classifier for each data set, with the design of each classifier termed a "task". One may learn separate classifiers for each of the J data sets (single-task learning) by ignoring connections between the data sets, or a single classifier may be learned based on the union of all data (pooling) by ignoring differences between the data sets. More appropriately, in a hierarchical Bayesian framework J task-dependent classifiers may be learned jointly, with information borrowed via a higher-level prior (multi-task learning). In some previous research all tasks are assumed to be equally related to each other (Yu et al., 2003; Zhang et al., 2006), or related tasks share exactly the same task-dependent classifier (Xue et al., 2007). With multiple local experts, the proposed iQGME model for a particular task is relatively flexible, enabling the borrowing of information across the J tasks (two data sets may share *parts* of the respective classifiers, without requiring sharing of all classifier components).

As discussed in Section 2.2, a DP prior encourages clustering (each cluster corresponds to a local expert). Now considering multiple tasks, a hierarchical Dirichlet process (HDP) (Teh et al., 2006) may be considered to solve the problem of sharing clusters (local experts) across multiple tasks. Assume a random measure G_j is associated with each task j, where each G_j is an independent draw from Dirichlet process $\mathcal{DP}(\alpha G_0)$ with a base measure G_0 drawn from an upper-level Dirichlet process $\mathcal{DP}(\beta H)$, that is,

$$\begin{aligned} G_j &\sim \mathcal{DP}(\alpha G_0), \quad \text{for} \quad j = 1, \ldots, J, \\ G_0 &\sim \mathcal{DP}(\beta H). \end{aligned}$$

As a draw from a Dirichlet process, G_0 is discrete with probability one and has a stick-breaking representation as in (7). With such a base measure, the task-dependent DPs reuse the atoms $\boldsymbol{\theta}_h^*$ defined in G_0, yielding the desired sharing of atoms among tasks.

With the task-dependent iQGME defined in (8), we consider all J tasks jointly:

$$\begin{aligned} (\boldsymbol{x}_{ji}, t_{ji}) &\sim \mathcal{N}_P(\boldsymbol{x}_{ji}|\boldsymbol{\mu}_{ji}, \boldsymbol{\Lambda}_{ji}^{-1})\mathcal{N}(t_{ji}|\boldsymbol{w}_{ji}^T\boldsymbol{x}_{ji}^b, 1), \\ (\boldsymbol{\mu}_{ji}, \boldsymbol{\Lambda}_{ji}, \boldsymbol{w}_{ji}) &\overset{iid}{\sim} G_j, \\ G_j &\sim \mathcal{DP}(\alpha G_0), \\ G_0 &\sim \mathcal{DP}(\beta H). \end{aligned}$$

In this form of borrowing information, experts with associated means and precision matrices are shared across tasks as distinct atoms. Since means and precision matrices statistically define local regions in feature space, sharing is encouraged locally. We explicitly write the stick-breaking representations for G_j and G_0, with z_{ji} and c_{jh} introduced as the indicators for each data point and each distinct atom of G_j, respectively. By factorizing the base measure H as a product of a normal-Wishart prior for $(\boldsymbol{\mu}_s, \boldsymbol{\Lambda}_s)$ and a normal prior for $\boldsymbol{w}_s$, the hierarchical model of the multi-task iQGME via the HDP is represented as

$$
\begin{aligned}
&\text{Data Generation:}\\
(t_{ji}|c_{jh}=s,z_{ji}=h) &\sim \mathcal{N}(\boldsymbol{w}_s^T\boldsymbol{x}_{ji}^b,1),\\
(\boldsymbol{x}_{ji}|c_{jh}=s,z_{ji}=h) &\sim \mathcal{N}_P(\boldsymbol{\mu}_s,\boldsymbol{\Lambda}_s^{-1}),\\
&\text{Drawing lower-level indicators:}\\
z_{ji} &\sim \sum_{h=1}^{\infty}\pi_{jh}\delta_h, \quad \text{where} \quad \pi_{jh}=V_{jh}\prod_{l<h}(1-V_{jl}),\\
V_{jh} &\sim Be(1,\alpha),\\
&\text{Drawing upper-level indicators:}\\
c_{jh} &\sim \sum_{s=1}^{\infty}\eta_s\delta_s, \quad \text{where} \quad \eta_s=U_s\prod_{l<s}(1-U_l),\\
U_s &\sim Be(1,\beta),\\
&\text{Drawing parameters from } H:\\
(\boldsymbol{\mu}_s,\boldsymbol{\Lambda}_s) &\sim \mathcal{N}_P(\boldsymbol{\mu}_s|\boldsymbol{m}_0,u_0^{-1}\boldsymbol{\Lambda}_s^{-1})\mathcal{W}(\boldsymbol{\Lambda}_s|\boldsymbol{B}_0,\nu_0),\\
\boldsymbol{w}_s &\sim \mathcal{N}_{P+1}(\boldsymbol{\zeta},[\text{diag}(\boldsymbol{\lambda})]^{-1}).
\end{aligned}
$$

where $j=1,\ldots,J$ and $i=1,\ldots,n_j$ index tasks and data points in each tasks, respectively; $h=1,\ldots,\infty$ and $s=1,\ldots,\infty$ index atoms for task-dependent G_j and the globally shared base G_0, respectively. Hyper-priors are imposed similarly as in the single-task case:

$$
\begin{aligned}
\alpha &\sim Ga(\tau_{10},\tau_{20}),\\
\beta &\sim Ga(\tau_{30},\tau_{40}),\\
(\boldsymbol{\zeta}|\boldsymbol{\lambda}) &\sim \mathcal{N}_{P+1}(\boldsymbol{0},\gamma_0^{-1}[\text{diag}(\boldsymbol{\lambda})]^{-1}),\\
\lambda_p &\sim Ga(a_0,b_0), \quad p=1,\ldots,P+1,
\end{aligned}
$$

The graphical representation of the iQGME for multi-task learning via the HDP is shown in Figure 2.

5. Variational Bayesian Inference

We initially present the inference formalism for single-task learning, and then discuss the (relatively modest) extensions required for the multi-task case.

5.1 Basic Construction

For simplicity we denote the collection of hidden variables and model parameters as Θ and specified hyper-parameters as Ψ. In a Bayesian framework we are interested in $p(\Theta|\mathcal{D},\Psi)$, the joint posterior distribution of the unknowns given observed data and hyper-parameters. From Bayes' rule,

$$p(\Theta|\mathcal{D},\Psi)=\frac{p(\mathcal{D}|\Theta)p(\Theta|\Psi)}{p(\mathcal{D}|\Psi)},$$

where $p(\mathcal{D}|\Psi)=\int p(\mathcal{D}|\Theta)p(\Theta|\Psi)d\Theta$ is the marginal likelihood that often involves multi-dimensional integrals. Since these integrals are nonanalytical in most cases, the computation of the marginal likelihood is the principal challenge in Bayesian inference. These integrals are circumvented if only a

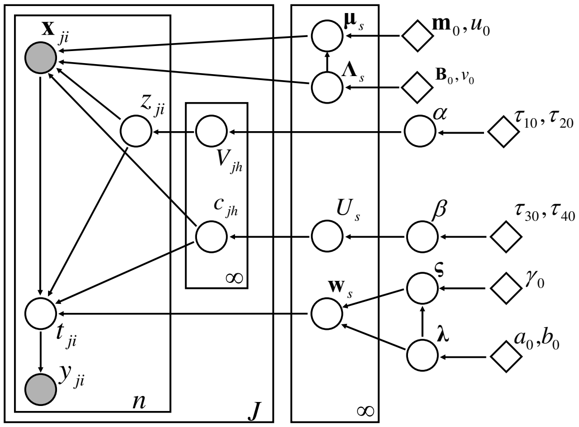


Figure 2: Graphical representation of the iQGME for multi-task leaning via the hierarchical Dirichlet process (HDP). Refer to Figure 1 for additional information.

point estimate $\hat{\Theta}$ is pursued, as in the expectation-maximization algorithm (Dempster et al., 1977). Markov chain Monte Carlo (MCMC) sampling methods (Gelfand et al., 1990; Neal, 1993) provide one class of approximations for the full posterior, based on samples from a Markov chain whose stationary distribution is the posterior of interest. As a Markov chain is guaranteed to converge to its true posterior theoretically as long as the chain is long enough, MCMC samples constitute an unbiased estimation for the posterior. Most previous applications with a Dirichlet process prior (Ishwaran and James, 2001; West et al., 1994), including the related papers we reviewed in Section 1, have been implemented with various MCMC methods. The main concerns of MCMC methods are associated with computational costs for computation of sufficient collection samples, and that diagnosis of convergence is often difficult.

As an efficient alternative, the variational Bayesian (VB) method (Beal, 2003) approximates the true posterior $p(\Theta|\mathcal{D},\Psi)$ with a variational distribution $q(\Theta)$ with free variational parameters. The problem of computing the posterior is reformulated as an optimization problem of minimizing the Kullback-Leibler (KL) divergence between $q(\Theta)$ and $p(\Theta|\mathcal{D},\Psi)$, which is equivalent to maximizing a lower bound of log $p(\mathcal{D}|\Psi)$, the log marginal likelihood. This optimization problem can be solved iteratively with two assumptions on $q(\Theta)$: (*i*) $q(\Theta)$ is factorized; (*ii*) the factorized components of $q(\Theta)$ come from the same exponential family as the corresponding priors do. Since the lower bound cannot achieve the true log marginal likelihood in general, the approximation given by the variational Bayesian method is biased. Another issue concerning the VB algorithm is that the solution may be trapped at local optima since the optimization problem is not convex. The main advantages of VB include the ease of convergence diagnosis and computational efficiency. As the VB is solving an optimization problem, the objective function—the lower bound of the log marginal likelihood—is a natural criterion for convergence diagnosis. Therefore, VB is a good alternative to MCMC when conjugacy is achieved and computational efficiency is desired. In recent publications (Blei and Jordan, 2006; Kurihara et al., 2007), discussions on the implementation of the variational Bayesian inference are given for Dirichlet process mixtures.

We implement the variational Bayesian inference throughout this paper, with comparisons made to Gibbs sampling. Since it is desirable to maintain the dependencies among random variables (e.g., shown in the graphical models Figure 1) in the variational distribution $q(\Theta)$, one typically only breaks those dependencies that bring difficulty to computation. In the subsequent inference for the iQGME, we retain some dependencies as unbroken. Following Blei and Jordan (2006), we employ stick-breaking representations with a truncation level N as variational distributions to approximate the infinite-dimensional random measures G.

We detail the variational Bayesian inference for the case of incomplete data. The inference for the complete-data case is similar, except that all feature vectors are fully observed and thus the step of learning missing values is skipped. To avoid repetition, a thorough procedure for the complete-data case is not included, with differences from the incomplete-data case indicated.

5.2 Single-task Learning

For single-task iQGME the unknowns are $\Theta = \{\boldsymbol{t}, \boldsymbol{x}^m, \boldsymbol{z}, \boldsymbol{V}, \alpha, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{W}, \boldsymbol{\zeta}, \boldsymbol{\lambda}\}$, with hyper-parameters $\Psi = \{\boldsymbol{m}_0, u_0, \boldsymbol{B}_0, \nu_0, \tau_{10}, \tau_{20}, \gamma_0, a_0, b_0\}$. We specify the factorized variational distributions as

$$\begin{aligned} & q(\boldsymbol{t}, \boldsymbol{x}^m, \boldsymbol{z}, \boldsymbol{V}, \alpha, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{W}, \boldsymbol{\zeta}, \boldsymbol{\lambda}) \\ = & \prod_{i=1}^{n}[q_{t_i}(t_i) q_{\boldsymbol{x}_i^{m_i}, z_i}(\boldsymbol{x}_i^{m_i}, z_i)] \prod_{h=1}^{N-1} q_{V_h}(V_h) \prod_{h=1}^{N}[q_{\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h}(\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h) q_{\boldsymbol{w}_h}(\boldsymbol{w}_h) \prod_{p=1}^{P+1} q_{\zeta_p, \lambda_p}(\zeta_p, \lambda_p) q_\alpha(\alpha) \end{aligned}$$

where

- $q_{t_i}(t_i)$ is a truncated normal distribution,

  $$t_i \sim \mathcal{TN}(\mu_i^t, 1, y_i t_i > 0), \quad i = 1, \ldots, n,$$

  which means the density function of t_i is assumed to be normal with mean μ_i^t and unit variance for those t_i satisfying $y_i t_i > 0$.

- $q_{\boldsymbol{x}_i^{m_i}, z_i}(\boldsymbol{x}_i^{m_i}, z_i) = q_{\boldsymbol{x}_i^{m_i}}(\boldsymbol{x}_i^{m_i} | z_i) q_{z_i}(z_i)$, where $q_{z_i}(z_i)$ is a multinomial distribution with probabilities $\boldsymbol{\rho}_i$, and there are N possible outcomes, $z_i \sim \mathcal{M}_N(1, \rho_{i1}, \ldots, \rho_{iN})$, $\quad i = 1, \ldots, n$. Given the associated indicators z_i, since features are assumed to be distributed as a multivariate Gaussian, the distributions of missing values $\boldsymbol{x}_i^{m_i}$ are still Gaussian according to conditional properties of multivariate Gaussian distributions:

  $$(\boldsymbol{x}_i^{m_i} | z_i = h) \sim \mathcal{N}_{|m_i|}(\boldsymbol{m}_h^{m_i|o_i}, \boldsymbol{\Sigma}_h^{m_i|o_i}), \quad i = 1, \ldots, n, \quad h = 1, \ldots, N.$$

  We retain the dependency between $\boldsymbol{x}_i^{m_i}$ and z_i in the variational distribution since the inference is still tractable; for complete data, the variation distribution for $(\boldsymbol{x}_i^{m_i} | z_i = h)$ is not necessary.

- $q_{V_h}(V_h)$ is a beta distribution,

  $$V_h \sim Be(v_{h1}, v_{h2}), \quad h = 1, \ldots, N-1.$$

  Recall that we have a truncation level of N, which implies that the mixture proportions $\pi_h(\boldsymbol{V})$ are equal to zero for $h > N$. Therefore, $q_{V_h}(V_h) = \delta_1$ for $h = N$, and $q_{V_h}(V_h) = \delta_0$ for $h > N$. For $h < N$, V_h has a variational Beta posterior.

- $q_{\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h}(\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h)$ is a normal-Wishart distribution,

  $$(\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h) \sim \mathcal{N}_P(\boldsymbol{m}_h, u_h^{-1} \boldsymbol{\Lambda}_h^{-1}) \mathcal{W}(\boldsymbol{B}_h, \nu_h), \quad h = 1, \ldots, N.$$

- $q_{\boldsymbol{w}_h}(\boldsymbol{w}_h)$ is a normal distribution,

$$\boldsymbol{w}_h \sim \mathcal{N}_{P+1}(\boldsymbol{\mu}_h^w, \boldsymbol{\Sigma}_h^w), \quad h = 1, \ldots, N.$$

- $q_{\zeta_p,\lambda_p}(\zeta_p, \lambda_p)$ is a normal-gamma distribution,

$$(\zeta_p, \lambda_p) \sim \mathcal{N}(\phi_p, \gamma^{-1}\lambda_p^{-1}) Ga(a_p, b_p), \quad p = 1, \ldots, P+1.$$

- $q_\alpha(\alpha)$ is a Gamma distribution,

$$\alpha \sim Ga(\tau_1, \tau_2).$$

Given the specifications on the variational distributions, a mean-field variational algorithm (Beal, 2003) is developed for the iQGME model. All update equations and derivations for $q(\boldsymbol{x}_i^{m_i}, z_i)$ are included in the Appendix; similar derivations for other random variables are found elsewhere (Xue et al., 2007; Williams et al., 2007). Each variational parameter is re-estimated iteratively conditioned on the current estimate of the others until the lower bound of the log marginal likelihood converges. Although the algorithm yields a bound for any initialization of the variational parameters, different initializations may lead to different bounds. To alleviate this local-maxima problem, one may perform multiple independent runs with random initializations, and choose the run that produces the highest bound on the marginal likelihood. We will elaborate on our initializations in the experiment section.

For simplicity, we omit the subscripts on the variational distributions and henceforth use q to denote any variational distributions. In the following derivations and update equations, we use generic notation $\langle f \rangle_{q(\cdot)}$ to denote $\mathrm{E}_{q(\cdot)}[f]$, the expectation of a function f with respect to variational distributions $q(\cdot)$. The subscript $q(\cdot)$ is dropped when it shares the same arguments with f.

5.3 Multi-task Learning

For multi-task learning much of the inference is highly related to that of single-task learning, as discussed above; in the following we focus only on differences. In the multi-task learning model, the latent variables are $\boldsymbol{\Theta} = \{\boldsymbol{t}, \boldsymbol{x}^m, \boldsymbol{z}, \boldsymbol{V}, \alpha, \boldsymbol{c}, \boldsymbol{U}, \beta, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{W}, \boldsymbol{\zeta}, \boldsymbol{\lambda}\}$, and hyper-parameters are $\boldsymbol{\Psi} = \{\boldsymbol{m}_0, u_0, \boldsymbol{B}_0, \nu_0, \tau_{10}, \tau_{20}, \tau_{30}, \tau_{40}, \gamma_0, a_0, b_0\}$. We specify the factorized variational distributions as

$$\begin{aligned} & q(\boldsymbol{t}, \boldsymbol{x}^m, \boldsymbol{z}, \boldsymbol{V}, \alpha, \boldsymbol{c}, \boldsymbol{U}, \beta, \boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{W}, \boldsymbol{\zeta}, \boldsymbol{\lambda}) \\ = \; & \prod_{j=1}^{J} \{ \prod_{i=1}^{n_j} [q(t_{ji}) q(\boldsymbol{x}_{ji}^{m_{ji}}) q(z_{ji})] \prod_{h=1}^{N-1} q(V_{jh}) \prod_{h=1}^{N} q(c_{jh}) \} \prod_{s=1}^{S-1} q(U_s) \\ & \prod_{s=1}^{S} [q(\boldsymbol{\mu}_s, \boldsymbol{\Lambda}_s) q(\boldsymbol{w}_s)] \prod_{p=1}^{P+1} q(\zeta_p, \lambda_p) q(\alpha) q(\beta) \end{aligned}$$

where the variational distributions of $(t_{ji}, V_{jh}, \alpha, \boldsymbol{\mu}_s, \boldsymbol{\Lambda}_s, \boldsymbol{w}_s, \zeta_p, \lambda_p)$ are assumed to be the same as in the single-task learning, while the variational distributions of hidden variables newly introduced for the upper-level Dirichlet process are specified as

- $q(c_{jh})$ for each indicator c_{jh} is a multinomial distribution with probabilities $\boldsymbol{\sigma}_{jh}$,

$$c_{jh} \sim \mathcal{M}_S(1, \sigma_{jh1}, \ldots, \sigma_{jhS}), \quad j = 1, \ldots, J, \quad h = 1, \ldots, N.$$

- $q(U_s)$ for each weight U_s is a Beta distribution,

  $$U_s \sim Be(\kappa_{s1}, \kappa_{s2}), \quad s = 1, \ldots, S-1.$$

  Here we have a truncation level of S for the upper-lever DP, which implies that the mixture proportions $\eta_s(\boldsymbol{U})$ are equal to zero for $s > S$. Therefore, $q(U_s) = \delta_1$ for $s = S$, and $q(U_s) = \delta_0$ for $s > S$. For $s < S$, U_s has a variational Beta posterior.

- $q(\beta)$ for the scaling parameter β is a Gamma distribution,

  $$\beta \sim Ga(\tau_3, \tau_4).$$

We also note that with a higher-level of hierarchy, the dependency between the missing values $\boldsymbol{x}_{ji}^{m_{ji}}$ and the associated indicator z_{ji} has to be broken so that the inference becomes tractable. The variational distribution of z_{ji} is still assumed to be multinomial distributed, while $\boldsymbol{x}_{ji}^{m_{ji}}$ is assumed to be normally distributed but no longer dependent on z_{ji}. All update equations are included in the Appendix.

5.4 Prediction

For a new observed feature vector $\boldsymbol{x}_\star^{o_\star}$, the prediction on the associated class label $y_\star$ is given by integrating out the missing values.

$$\begin{aligned} P(y_\star = 1 | \boldsymbol{x}_\star^{o_\star}, \mathcal{D}) = \frac{p(y_\star = 1, \boldsymbol{x}_\star^{o_\star} | \mathcal{D})}{p(\boldsymbol{x}_\star^{o_\star} | \mathcal{D})} &= \frac{\int p(y_\star = 1, \boldsymbol{x}_\star | \mathcal{D}) d\boldsymbol{x}_\star^{m_\star}}{\int p(\boldsymbol{x}_\star | \mathcal{D}) d\boldsymbol{x}_\star^{m_\star}} \\ = \quad & \frac{\int \sum_{h=1}^N P(z_\star = h | \mathcal{D}) p(x_\star | z_\star = h, \mathcal{D}) P(y_\star = 1 | \boldsymbol{x}_\star, z_\star = h, \mathcal{D}) d\boldsymbol{x}_\star^{m_\star}}{\int \sum_{k=1}^N P(z_\star = k | \mathcal{D}) p(x_\star | z_\star = k, \mathcal{D}) d\boldsymbol{x}_\star^{m_\star}}. \end{aligned}$$

We marginalize the hidden variables over their variational distributions to compute the predictive probability of the class label

$$P(y_\star = 1 | \boldsymbol{x}_\star^{o_\star}, \mathcal{D}) = \frac{\sum_{h=1}^N \mathrm{E}_V[\pi_h] \int_0^\infty \int \mathrm{E}_{\boldsymbol{\mu}_h \boldsymbol{\Lambda}_h} [\mathcal{N}_P(\boldsymbol{x}_\star | \boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h^{-1})] \mathrm{E}_{\boldsymbol{w}_h} \left[\mathcal{N}(t_\star | \boldsymbol{w}_h^T \boldsymbol{x}_\star^b, 1) \right] d\boldsymbol{x}_\star^{m_\star} dt_\star}{\sum_{k=1}^N \mathrm{E}_V[\pi_k] \int \mathrm{E}_{\boldsymbol{\mu}_k \boldsymbol{\Lambda}_k} [\mathcal{N}_P(\boldsymbol{x}_\star | \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})] d\boldsymbol{x}_\star^{m_\star}}$$

where

$$\mathrm{E}_V[\pi_h] = \mathrm{E}_V[V_h \prod_{l<h} (1 - V_l)] = \langle V_h \rangle \prod_{l<h} \langle 1 - V_l \rangle = \left[\frac{\nu_{h1}}{\nu_{h1} + \nu_{h2}} \right]^{\mathbf{1}(h<N)} \prod_{l<h} \left[\frac{\nu_{l2}}{\nu_{l1} + \nu_{l2}} \right]^{\mathbf{1}(h>1)}.$$

The expectation $\mathrm{E}_{\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h} [\mathcal{N}_P(\boldsymbol{x}^\star | \boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h^{-1})]$ is a multivariate Student-t distribution (Attias, 2000). However, for the incomplete-data situation, the integral over the missing values is tractable only when the two terms in the integral are both normal. To retain the form of norm distributions, we use the posterior means of $\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h$ and $\boldsymbol{w}_h$ to approximate the variables:

$$\begin{aligned} P(y_\star = 1 | \boldsymbol{x}_\star^{o_\star}, \mathcal{D}) \quad &\approx \quad \frac{\sum_{h=1}^N \mathrm{E}_V[\pi_h] \int_0^\infty \int \mathcal{N}_P(\boldsymbol{x}_\star | \boldsymbol{m}_h, (\nu_h \boldsymbol{B}_h)^{-1}) \mathcal{N}(t_\star | (\boldsymbol{\mu}_h^w)^T \boldsymbol{x}_\star^b, 1) d\boldsymbol{x}_\star^{m_\star} dt_\star}{\sum_{k=1}^N \mathrm{E}_V[\pi_k] \int \mathcal{N}_P(\boldsymbol{x}_\star | \boldsymbol{m}_k, (\nu_k \boldsymbol{B}_k)^{-1}) d\boldsymbol{x}_\star^{m_\star}} \\ &= \quad \frac{\sum_{h=1}^N \mathrm{E}_V[\pi_h] \mathcal{N}_{|o_\star|}(\boldsymbol{x}_\star^{o_\star} | \boldsymbol{m}_h^{o_\star}, (\nu_h \boldsymbol{B}_h)^{-1, o_\star o_\star}) \int_0^\infty \mathcal{N}(t_\star | \varphi_{\star h}, g_{\star h}) dt_\star}{\sum_{k=1}^N \mathrm{E}_V[\pi_k] \mathcal{N}_{|o_\star|}(\boldsymbol{x}_\star^{o_\star} | \boldsymbol{m}_k^{o_\star}, (\nu_k \boldsymbol{B}_k)^{-1, o_\star o_\star})} \end{aligned}$$

where

$$\begin{aligned}
\varphi_{\star h} &= [\boldsymbol{m}_h^T, 1]\boldsymbol{\mu}_h^w + \boldsymbol{\Gamma}_{\star h}^T (\boldsymbol{\Delta}_h^{o_\star o_\star})^{-1} (\boldsymbol{x}_\star^{o_\star} - \boldsymbol{m}_h^{o_\star}), \\
g_{\star h} &= 1 + (\bar{\boldsymbol{\mu}}_h^w)^T \boldsymbol{\Delta}_h \bar{\boldsymbol{\mu}}_h^w - \boldsymbol{\Gamma}_{\star h}^T (\boldsymbol{\Delta}_h^{o_\star o_\star})^{-1} \boldsymbol{\Gamma}_{\star h}, \\
\boldsymbol{\Gamma}_{\star h} &= \boldsymbol{\Delta}_h^{o_\star o_\star} (\boldsymbol{\mu}_h^w)^{o_\star} + \boldsymbol{\Delta}_h^{o_\star m_\star} (\boldsymbol{\mu}_h^w)^{m_\star}, \\
\bar{\boldsymbol{\mu}}_h^w &= (\boldsymbol{\mu}_h^w)_{1:P}, \\
\boldsymbol{\Delta}_h &= (\nu_h \boldsymbol{B}_h)^{-1}.
\end{aligned}$$

For complete data the integral of missing features is absent, so we take advantage of the full variational posteriors for prediction.

5.5 Computational Complexity

Given the truncation level (or the number of clusters) N, the data dimensionality P, and the number of data points n, we compare the iQGME to closely related DP regression models (Meeds and Osindero, 2006; Shahbaba and Neal, 2009), in terms of the time and memory complexity. The inference of the iQGME with complete data requires inversion of two $P \times P$ matrices (the covariance matrices for the inputs and the local expert) associated with each cluster. Therefore, the time and memory complexity are $O(2NP^3)$ and $O(2NP^2)$, respectively. With incomplete data, since the missing pattern is unique for each data point, the time and memory complexity increase with number of data points, that is, $O(nNP^3)$ and $O(nNP^2)$, respectively. The mixture of Gaussian process experts (Meeds and Osindero, 2006) requires $O(NP^3 + n^3/N)$ computations for each MCMC iteration if the N experts equally divide the data, and the memory complexity is $O(NP^2 + n^2/N)$. In the model proposed by Shahbaba and Neal (2009), no matrix inversion is needed since the covariates are assumed to be independent. The time and memory complexity are $O(NP)$ and $O(NP)$, respectively.

From the aspect of computational complexity, the model in Meeds and Osindero (2006) is restricted by the increase of both dimensionality and data size; while the model proposed in Shahbaba and Neal (2009) is more efficient. Although the proposed model requires more computations for each MCMC iteration than the latter one, we are able to handle missing values naturally, and much more efficiently compared to the former one. Considering the usual number of iterations required by VB (several dozens) and MCMC (thousands or even tens of thousands), our model is even more efficient.

6. Experimental Results

In all the following experiments the hyper-parameters are set as follows: $a_0 = 0.01$, $b_0 = 0.01$, $\gamma_0 = 0.1$, $\tau_{10} = 0.05$, $\tau_{20} = 0.05$, $\tau_{30} = 0.05$, $\tau_{40} = 0.05$, $u_0 = 0.1$, $\nu_0 = P + 2$, and $\boldsymbol{m}_0$ and $\boldsymbol{B}_0$ are set according to sample mean and sample precision, respectively. These parameters have not been optimized for any particular data set (which are all different in form), and the results are relatively insensitive to "reasonable" settings. The truncation levels for the variational distributions are set to be $N = 20$ and $S = 50$. We have found the results insensitive to the truncation level, for values larger than those considered here.

Because of the local-maxima issue associated with VB, initialization of the inferred VB hyper-parameters is often important. We initialize most variational hyper-parameters using the corresponding prior hyper-parameters, which are data-independent. The precision/covariance matrices

$\boldsymbol{B}_h$ and $\boldsymbol{\Sigma}_h^w$ are simply initialized as identity matrices. However, for several other hyper-parameters, we may obtain information for good start points from the data. Specifically, the variational mean of the soft label μ_i^t is initialized by the associated label y_i. A K-means clustering algorithm is implemented on the feature vectors, and the cluster means and identifications for objects are used to initialize the variational mean of the Gaussian means $\boldsymbol{m}_h$ and the indicator probabilities $\boldsymbol{\rho}_i$, respectively. As an alternative, one may randomly initialize $\boldsymbol{m}_h$ and $\boldsymbol{\rho}_i$ multiple times, and select the solution that produces the highest lower bound on the log marginal likelihood. The two approaches work almost equivalently for low-dimensional problems; however, for problems with moderate to high dimensionality, it could be fairly difficult to get a satisfying initialization by making several random trials.

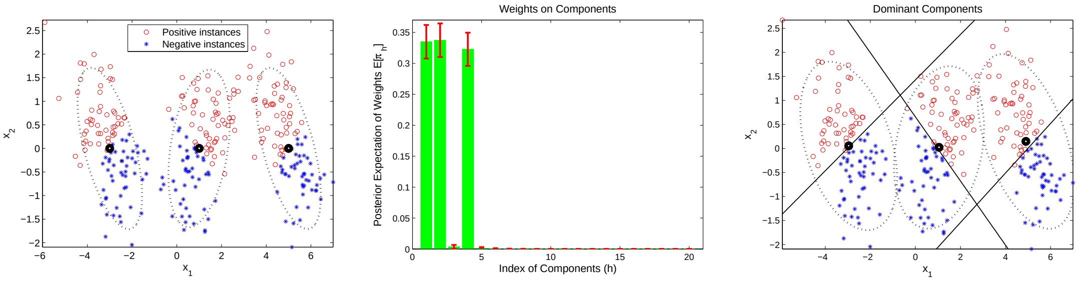
6.1 Synthetic Data

We first demonstrate the proposed iQGME single-task learning model on a synthetic data set, for illustrative purposes. The data are generated according to a GMM model $p(\boldsymbol{x}) = \sum_{k=1}^{3} \pi_k \mathcal{N}_2(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ with the following parameters:

$$\boldsymbol{\pi} = \left[\begin{array}{ccc} 1/3 & 1/3 & 1/3 \end{array}\right], \quad \boldsymbol{\mu}_1 = \left[\begin{array}{c} -3 \\ 0 \end{array}\right], \quad \boldsymbol{\mu}_2 = \left[\begin{array}{c} 1 \\ 0 \end{array}\right], \quad \boldsymbol{\mu}_3 = \left[\begin{array}{c} 5 \\ 0 \end{array}\right]$$

$$\boldsymbol{\Sigma}_1 = \left[\begin{array}{cc} 0.52 & -0.36 \\ -0.36 & 0.73 \end{array}\right], \quad \boldsymbol{\Sigma}_2 = \left[\begin{array}{cc} 0.47 & 0.19 \\ 0.19 & 0.7 \end{array}\right], \quad \boldsymbol{\Sigma}_3 = \left[\begin{array}{cc} 0.52 & -0.36 \\ -0.36 & 0.73 \end{array}\right].$$

The class boundary for each Gaussian component is given by three lines $x_2 = w_k x_1 + b_k$ for $k = 1,2,3$, where $w_1 = 0.75, b_1 = 2.25$, $w_2 = -0.58, b_2 = 0.58$, and $w_3 = 0.75, b_3 = -3.75$. The simulated data are shown in Figure 3(a), where black dots and dashed ellipses represent the true means and covariance matrices of the Gaussian components, respectively.



(a) (b) (c)

Figure 3: Synthetic three-Gaussian single-task data with inferred components. (a) Data in feature space with true labels and true Gaussian components indicated; (b) inferred posterior expectation of weights on components, with standard deviations depicted as error bars; (c) ground truth with posterior means of dominant components indicated (the linear classifiers and Gaussian ellipses are inferred from the data).

The inferred mean mixture weights with standard deviations are depicted in Figure 3(b), and it is observed that three dominant mixture components (local "experts") are inferred. The dominant components (those with mean weight larger than 0.005) are characterized by Gaussian means, covariance matrices and local experts, as depicted in Figure 3(c). From Figure 3(c), the nonlinear classification is manifested by using three dominant *local* linear classifiers, with a GMM defining the effective regions stochastically.

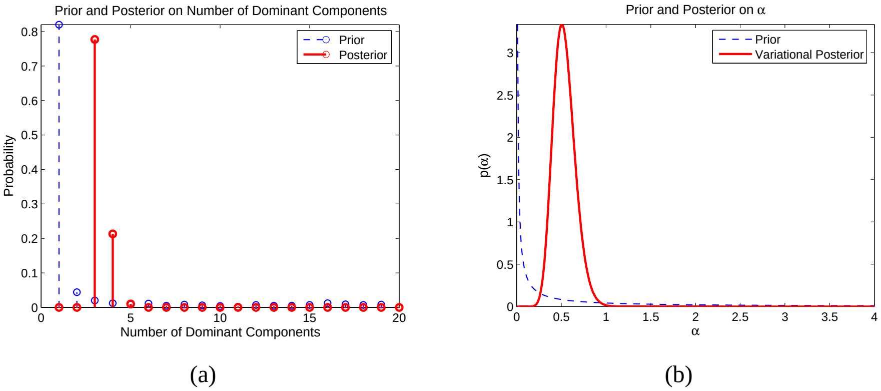


Figure 4: Synthetic three-Gaussian single-task data: (a) prior and posterior beliefs on the number of dominant components; (b) prior and posterior beliefs on α.

An important point is that we are not selecting a “correct” number of mixture components as in most mixture-of-expert models, including the finite QGME model (Liao et al., 2007). Instead, there exists uncertainty on the number of components in our posterior belief. Since this uncertainty is not inferred directly, we obtain samples for the number of dominant components by calculating π_h based on V_h sampled from their probability density functions (prior or variational posterior), and the probability mass functions given by histogram are shown in Figure 4(a). As discussed, the scale parameter α is highly related to the number of clusters, so we depict the prior and the variational posterior on α in Figure 4(b).

The predictions in feature space are presented in Figure 5, where the prediction in sub-figure (a) is given by integrating over the full posteriors of local experts and parameters (means and covariance matrices) of Gaussian components; while the prediction in sub-figure (b) is given by posterior means. We examine these two cases since the analytical integrals over the full posteriors may be unavailable sometimes in practice (for example, for cases with incomplete data as discussed in Section 5). From Figures 5(a) and 5(b), we observe that these two predictions are fairly similar, except that (a) allows more uncertainty on regions with scarce data. The reason for this is that the posteriors are often peaked and thus posterior means are usually representative. As an example, we plot the broad common prior imposed for local experts in Figure 5(c) and the peaked variational posteriors for three dominant experts in Figure 5(d). According to Figure 5, we suggest the usage of full posteriors for prediction whenever integrals are analytical, that is, for experiments with complete data. It also empirically justifies the use of posterior means as an approximation. These results have been computed using VB inference, with MCMC-based results presented below, as a comparison.

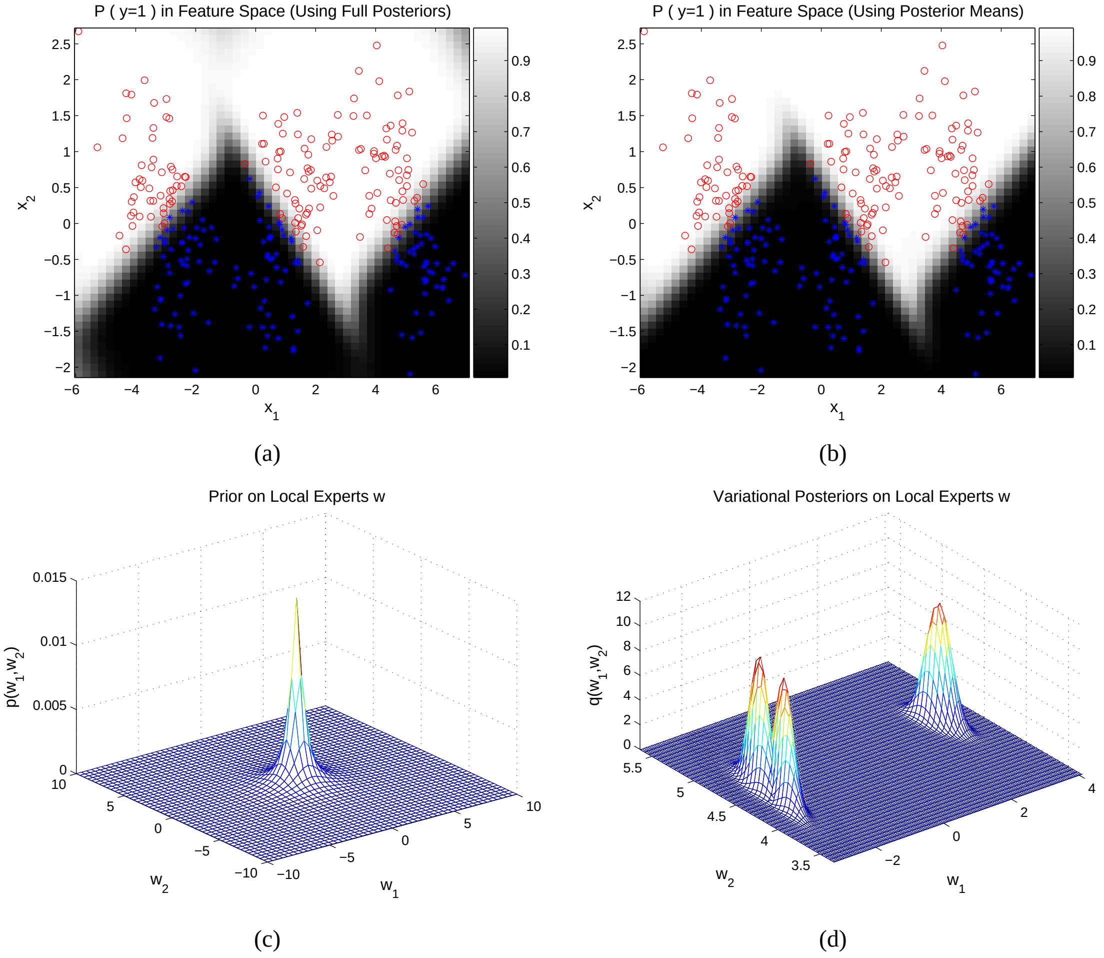


Figure 5: Synthetic three-Gauss single-task data: (a) prediction in feature space using the full posteriors; (b) prediction in feature space using the posterior means; (c) a common broad prior on local experts; (d) variational posteriors on local experts.

6.2 Benchmark Data

To further evaluate the proposed iQGME, we compare it with other models, using benchmark data sets available from the UCI machine learning repository (Newman et al., 1998). Specifically, we consider Wisconsin Diagnostic Breast Cancer (WDBC) and the Johns Hopkins University Ionosphere database (Ionosphere) data sets, which have been studied in the literature (Williams et al., 2007; Liao et al., 2007). These two data sets are summarized in Table 1.

The models we compare to include:

- State-of-the-art classification algorithms: Support Vector Machines (SVM) (Vapnik, 1995) and Relevance Vector Machines (RVM) (Tipping, 2000). We consider both linear models (Linear) and non-linear models with radial basis function (RBF) for both algorithms. For

Data set	Dimension	Number of positive instances	Number of negative instances
Ionosphere	34	126	225
WDBC	30	212	357

Table 1: Details of Ionosphere and WDBC data sets

each data set, the kernel parameter is selected for one training/test/validation separation, and then fixed for all the other experimental settings. The RVM models are implemented with Tipping's Matlab code available at `http://www.miketipping.com/index.php?page=rvm`.

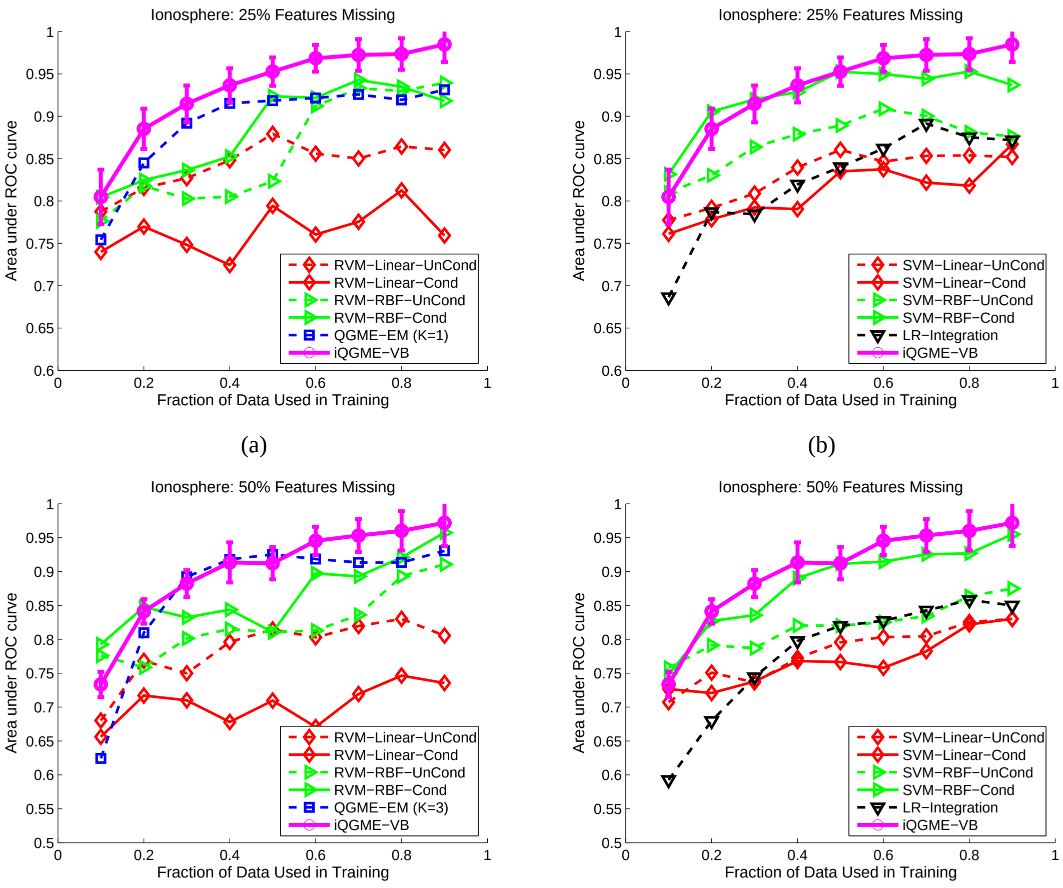
Since those SVM and RVM algorithms are not directly applicable to problems with missing features, we use two methods to impute the missing values before the implementation. One is using the mean of observed values (unconditional mean) for the given feature, referred to as Uncond; the other is using the posterior mean conditional on observed features (conditional mean), referred to as Cond (Schafer and Graham, 2002).

- Classifiers handling missing values: the finite QGME inferred by expectation-maximization (EM) (Liao et al., 2007), referred to as QGME-EM, and a two-stage algorithm (Williams et al., 2007) where the parameters of the GMM for the covariates are estimated first given the observed features, and then a marginalized linear logistic regression (LR) classifier is learned, referred to as LR-Integration. Results are cited from Liao et al. (2007) and Williams et al. (2007), respectively.

In order to simulate the *missing at random* setting, we randomly remove a fraction of feature values according to a uniform distribution, and assume the rest are observed. Any instance with all feature values missing is deleted. After that, we randomly split each data set into training and test subsets, imposing that each subset encompasses at least one instance from each of the classes. Note that the random pattern of missing features and the random partition of training and test subsets are independent of each other. By performing multiple trials we consider the general (average) performance for various data settings. For convenient comparison with Williams et al. (2007) and Liao et al. (2007), the performance of algorithms is evaluated in terms of the area under a receiver operating characteristic (ROC) curve (AUC) (Hanley and McNeil, 1982).

The results on the Ionosphere and WDBC data sets are summarized in Figures 6 and 7, respectively, where we consider 25% and 50% of the feature values missing. Given a portion of missing values, each curve is a function of the fraction of data used in training. For a given size of training data, we perform ten independent trials for the SVM and RVM models and the proposed iQGME.

From both Figures 6 and 7, the proposed iQGME-VB is robust for all the experimental settings, and its overall performance is the best among all algorithms considered. Although the RVM-RBF-Cond and the SVM-RBF-Cond perform well for the Ionosphere data set, especially when the training data is limited, their performance on the WDBC data set is not as good. The kernel methods benefit from the introduction of the RBF kernel for the Ionosphere data set; however, the performance is inferior for the WDBC data set. We also note that the one-step iQGME and the finite QGME outperform the two-step LR-integration. The proposed iQGME consistently performs better than the finite QGME (where, for the latter, in all cases we show results for the best/optimized choice of number of experts K), which reveals the advantage of retaining the uncertainty on the model structure (number of experts) and model parameters. As shown in Figure 7, the advantage of



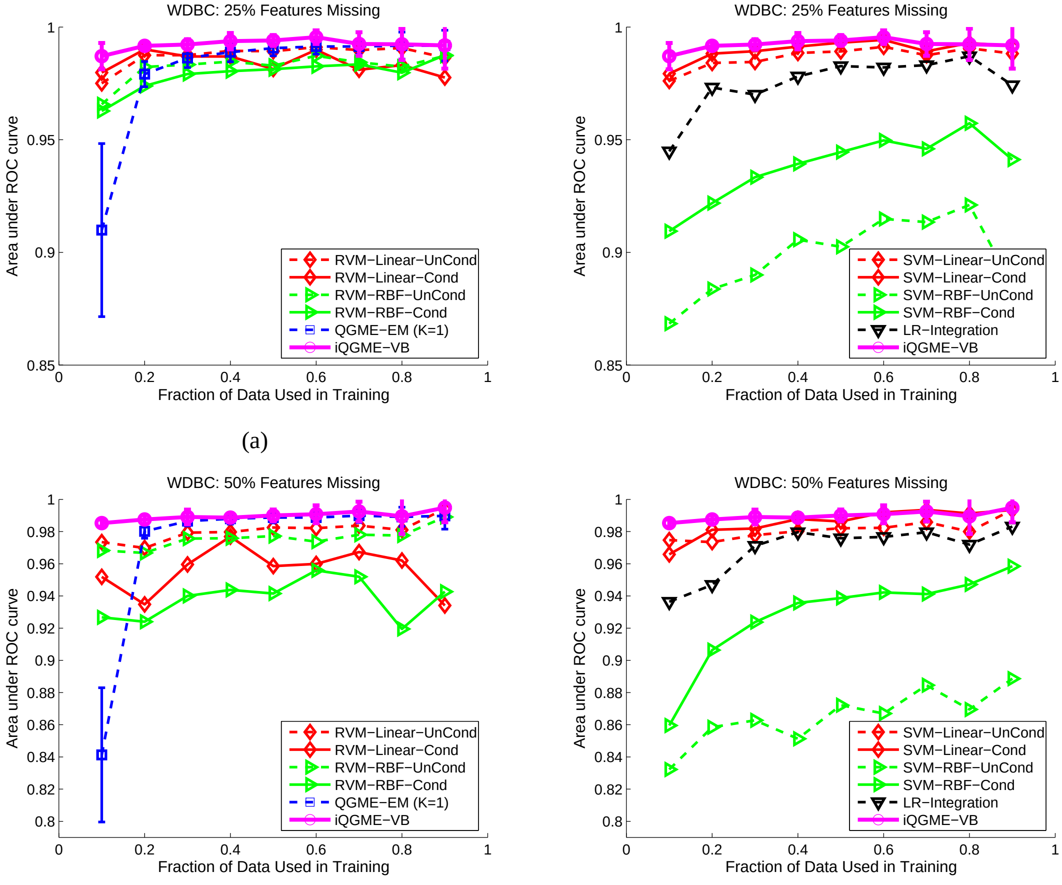
(a) (b)

(c) (d)

Figure 6: Results on Ionosphere data set for (a)(b) 25%, and (c)(d) 50% of the feature values missing. For legibility, we only report the standard deviation for the proposed iQGME-VB algorithm as error bars, and present the compared algorithms in two figures for each case. The results of the finite QGME solved with an expectation-maximization method are cited from Liao et al. (2007), and those of LR-Integration are cited from Williams et al. (2007). Since the performance of the QGME-EM is affected by the choice of number of experts K, the overall best results among $K = 1,3,5,10,20$ are cited for comparison in each case (no such selection of K is required for the proposed iQGME-VB algorithm).

considering the uncertainty on the model parameters is fairly pronounced for the WDBC data set, especially when training examples are relatively scarce and thus the point-estimation EM method suffers from over-fitting issues. A more detailed examination on the model uncertainty is shown in Figures 8 and 9.

In Figure 8, the influence of the preset value for K on the QGME-EM model is examined on the Ionosphere data. We observe that with different fractions of missing values and training samples, the values for K which achieve the best performance may be different; as K goes to a large number (e.g., 20 here), the performance gets worse due to over-fitting. In contrast, we do not need to set the



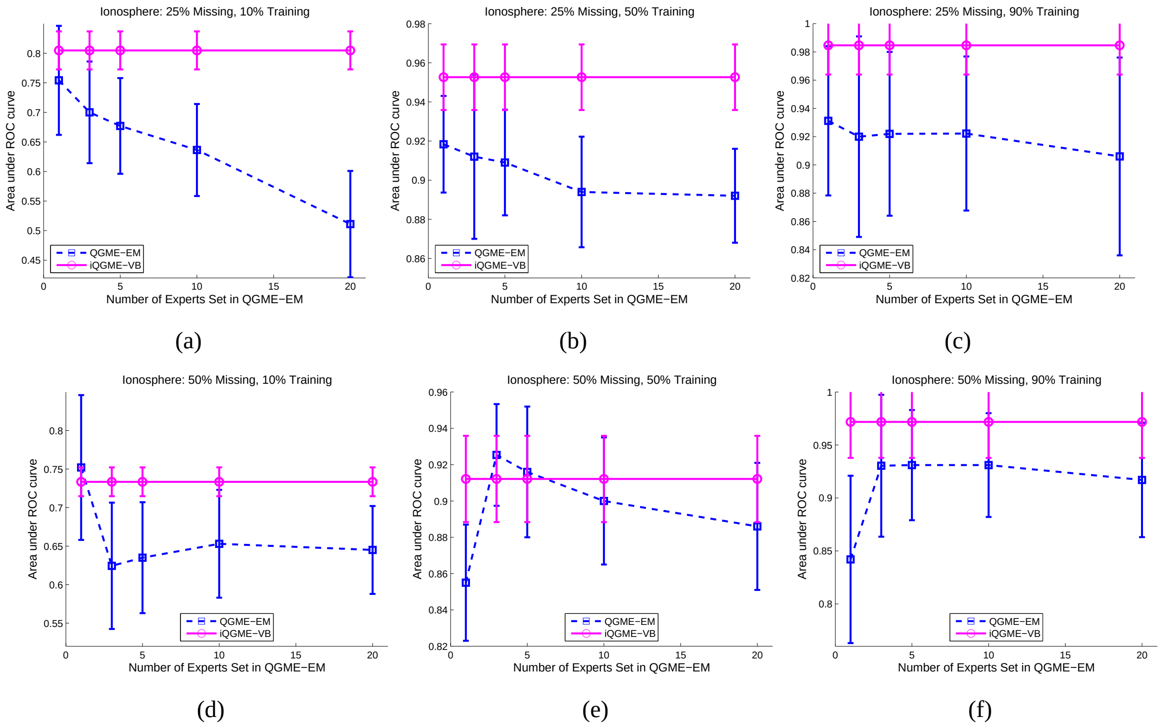
(a)

(c)

Figure 7: Results on WDBC data set for cases when (a)(b) 25%, and (c)(d) 50% of the feature values are missing. Refer to Figure 6 for additional information.

number of clusters for the proposed iQGME-VB model. As long as the truncation level N is large enough ($N = 20$ for all the experiments), the number of clusters is inferred by the algorithm. We give an example for the posterior on the number of clusters inferred by the proposed iQGME-VB model, and report the statistics for the most probable number of experts given each missing fraction and training fraction in Figure 9, which suggests that the number of clusters may vary significantly even for the trials with the same fraction of feature values missing and the same fraction of samples for training. Therefore, it may be not appropriate to set a fixed value for the number of clusters for all the experimental settings as one has to do for the QGME-EM.

Although our main purpose is classification, one may also be interested in how well the algorithm can estimate the missing values while pursuing the main purpose. In Figure 10, we show the ratio of correctly estimated missing values for the Ionosphere data set with 25% feature values missing, where two criteria are considered: true values are one standard deviation (red circles) or two standard deviations (blue squares) away from the posterior means. This figure suggests that the algorithm estimates most of the missing values in a reasonable range away from the true val-



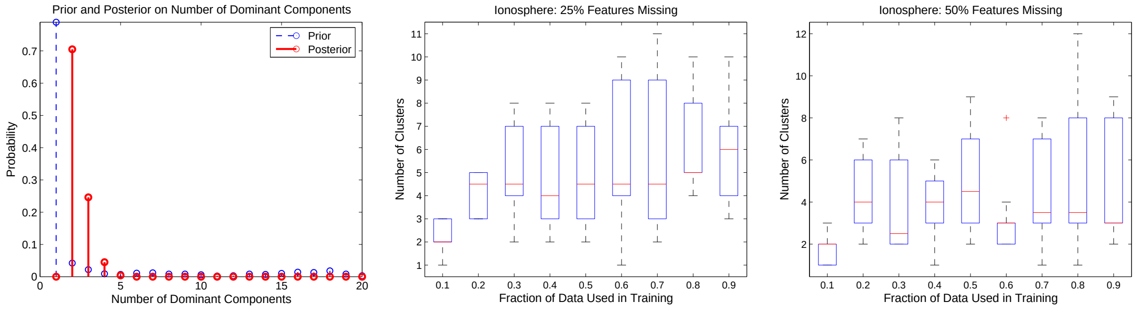
(a) (b) (c)

(d) (e) (f)

Figure 8: The comparison on the Ionosphere data set between QGME-EM with different preset number of clusters K and the proposed iQGME-VB, when (a)(b)(c) 25%, and (d)(e)(f) 50% of the features are missing. In each row, 10%, 50%, and 90% of samples are used for training, respectively. Results of QGME-EM are cited from Liao et al. (2007).

ues when the training size is large enough; even with not so satisfying estimations (as for limited training data), the classification results are still relatively robust as shown in Figure 6.

We have discussed the advantages and disadvantages for the inference with MCMC and VB in Section 5.1. Here we take the Ionosphere data with 25% features missing as an example to compare these two inference techniques, as shown in Figure 11. It can be seen that they achieve similar performance for the particular iQGME model proposed in this paper. The time consumed for each iteration is also comparable, and increases almost linearly with the training size, as discussed in Section 5.5. The VB inference takes a little bit longer per iteration, probably due to the extra computation for the lower bound of the log marginal likelihood, which serves as convergence criterion. Significant differences occur on the number of iterations we have to take. In the experiment, even though we set a very strict threshold (10^{-6}) for the relative change of the lower bound, the VB algorithm converges at about 50 iterations for most cases except when training data are very scarce (10%). For the MCMC inference, we discard the initial samples from the first 1000 iterations (burn-in), and collect the next 500 samples to present the posterior. It is far from enough to claim convergence; however, we consider it a fair comparison for computation as the two methods yield similar results under this setting. Given the fact that the VB algorithm only takes about $1/30$ the CPU time, and VB and MCMC performance are similar, in the following examples we only present results based on VB inference. However, in all the examples below we also performed Gibbs sam-



(a) (b) (c)

Figure 9: Number of clusters for the Ionosphere data set inferred by iQGME-VB. (a) Prior and inferred posterior on the number of clusters for one trial given 10% samples for training, the number of clusters for the case when (b) 25%, and (c) 50% of features are missing. The most probable value of clusters number is used for each trial to generate (b) and (c) (e.g., the most probable value of clusters number is two for the trial shown in (a)). In (b) and (c), the distribution of number of clusters for the ten trials given each missing fraction and training fraction is presented as a box-plot, where the red line represents the median; the bottom and top of the blue box are the 25th and 75th percentile, respectively; the bottom and top black lines are the end of the whiskers, which could be the minimum and maximum, respectively; if some data are beyond 1.5 times of the length of the blue box (interquartile range), they are outliers, indicated by a red '+'.

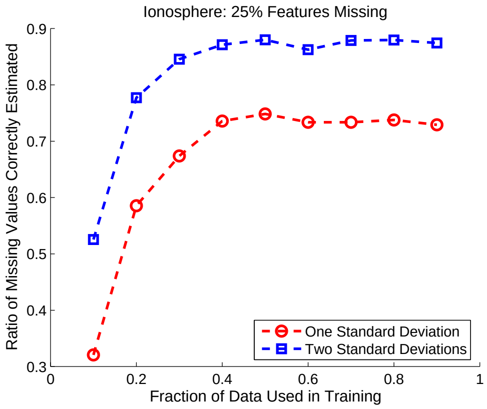
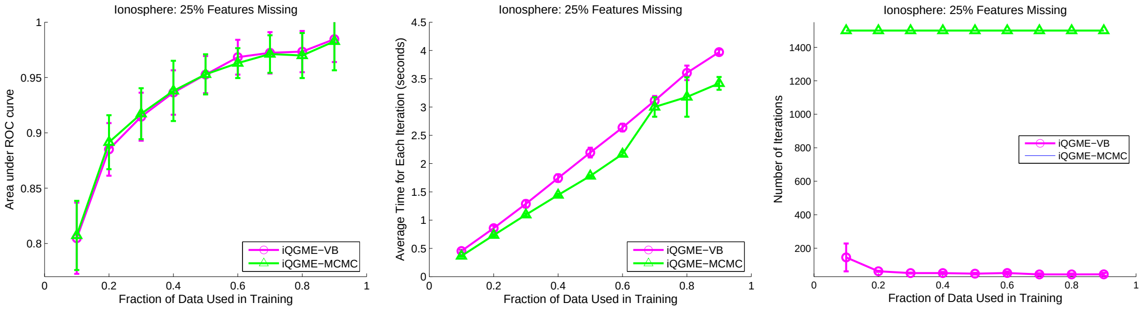


Figure 10: Ratio of missing values whose true values are one standard deviation (red circles) or two standard deviations (blue squares) away from the posterior means for the Ionosphere data set with 25% feature values missing. One trial for each training size is considered.

pling, and the relative inference consistency and computational costs relative to VB were found to be as summarized here (i.e., in all cases there was close agreement between the VB and MCMC inferences, and considerable computational acceleration manifested by VB).



(a) (b) (c)

Figure 11: Comparison between VB and MCMC inferred iQGME on the Ionosphere data with 25% features missing in terms of (a) performance, (b) time consumed for each iteration, (c) number of iterations. For the VB inference, we set a threshold (10^{-6}) for the relative change of lower bound in two consecutive iterations as the convergence criterion; for the MCMC inference, we discard the initial samples from the first 1000 iterations (burn-in), and collect the next 500 samples to present the posterior.

6.3 Unexploded Ordnance Data

We now consider an unexploded ordnance (UXO) detection problem (Zhang et al., 2003), where two types of sensors are used to collect data, but one of them may be absent for particular targets. Specifically, one sensor is a magnetometer (MAG) and the other an electromagnetic induction (EMI) sensor; these sensors are deployed separately to interrogate buried targets, and for some targets both sensors are deployed and for others only one sensor is deployed. This is a real sensing problem for which missing data occurs naturally. The data considered were made available to the authors by the US Army (and were collected from a real former bombing range in the US); the data are available to other researchers upon request. The total number of targets are 146, where 79 of them UXO and the rest are non-UXO (i.e., non-explosives). A six-dimensional feature vector is extracted from the raw signals to represent each target, with the first three components corresponding to MAG features and the rest as EMI features (details on feature extraction is provided in Zhang et al., 2003). Figure 12 shows the missing patterns for this data set.

We compare the proposed iQGME-VB algorithm with the SVM, RVM and LR-Integration as detailed in Section 6.2. In order to evaluate the overall performance of classifiers, we randomly partition the training and test subsets, and change the training size. Results are shown in Figure 13, where only performance means are reported for the legibility of the figures. From this figure, the proposed iQGME-VB method is robust for all the experimental settings under both performance criteria.

6.4 Sepsis Classification Data

In Sections 6.2 and 6.3, we have demonstrated the proposed iQGME-VB on data sets with low to moderate dimensionality. A high-dimensional data set with natural missing values is considered in this subsection. These data were made available to the authors from the National Center for Genomic Research in the US, and will be made available upon request. This is another example

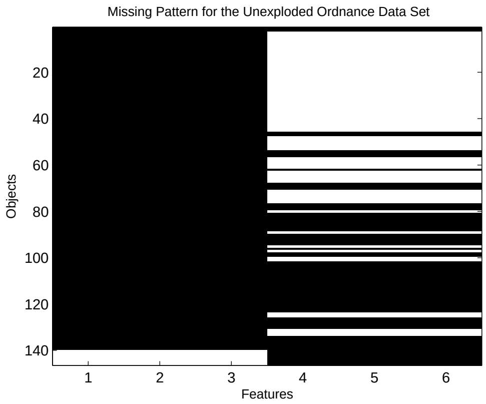
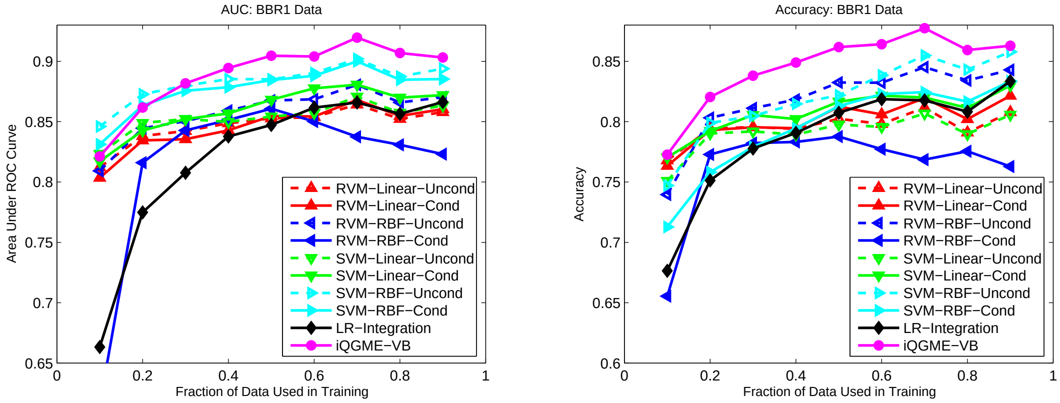


Figure 12: Missing pattern for the unexploded ordnance data set, where black and white indicate observed and missing, respectively.



(a)

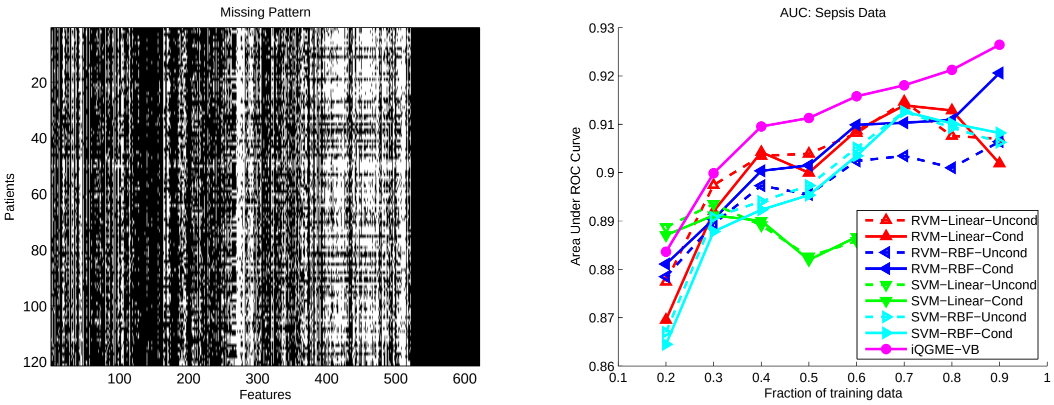
(b)

Figure 13: Mean performance over 100 random training/test partitions for each training fraction on the unexploded ordnance data set, in terms of (a) area under the ROC curve, and (b) classification accuracy.

for which missing data are a natural consequence of the sensing modality. There are 121 patients who are infected by sepsis, with 90 of them surviving (label -1) and 31 of them who die (label 1). For each patient, we have 521 metabolic features and 100 protein features. The purpose is to predict whether a patient infected by sepsis will die given his/her features. The missing pattern of feature values is shown in Figure 14(a), where black indicates observed (this missingness is a natural consequence of the sensing device).

As the data are in a 621-dimensional feature space, with only 121 samples available, we use the MFA-based variant of the iQGME (Section 2.4). To impose the low-rank belief for each cluster, we set $c_0 = d_0 = 1$, and the largest possible dimensionality for clusters is set to be $L = 50$.

We compare to the same algorithms considered in Section 6.3, except the LR-Integration algorithm since it is not capable of handling such a high-dimensional data set. Mean AUC over ten random partitions are reported in Figure 14(b). Here we report the SVM and RVM results on the original data since they are able to classify the data in the original 621-dimensional space after missing values are imputed; we also examined SVM and RVM results on the data in a lower-dimensional latent space, after first performing factor analysis on the data, and these results were very similar to the SVM/RVM results in the original 621-dimensional space. From Figure 14(b), our method provides improvement by handling missing values analytically in the procedure of model inference and performing a dimensionality reduction jointly with local classifiers learning.



(a)

(b)

Figure 14: Sepsis data set. (a) Missing pattern, where black and white indicate observed and missing, respectively, (b) mean performance over 100 random training/test partitions for each training fraction.

6.5 Multi-Task Learning with Landmine Detection Data

We now consider a multi-task-learning example. In an application of landmine detection (available at `http://www.ee.duke.edu/~lcarin/LandmineData.zip`), data collected from 19 landmine fields are treated as 19 subtasks. Among them, subtasks 1-10 correspond to regions that are relatively highly foliated and subtasks 11-19 correspond to regions that are bare earth or desert. In all subtasks, each target is characterized by a 9-dimensional feature vector $\boldsymbol{x}$ with corresponding binary label y (1 for landmines and -1 for clutter). The number of landmines and clutter in each task is summarized in Figure 15. The feature vectors are extracted from images measured with airborne radar systems. A detailed description of this landmine data set has been presented elsewhere (Xue et al., 2007).

Although our main objective is to simultaneously learn classifiers for multiple tasks with incomplete data, we first demonstrate the proposed iQGME-based multi-task learning (MTL) model on the complete data, comparing it to two multi-task learning algorithms designed for the situation with all the features observed. One is based on task-specific logistic regression (LR) models, with the DP as a hierarchical prior across all the tasks (Xue et al., 2007); the other assumes an underlying

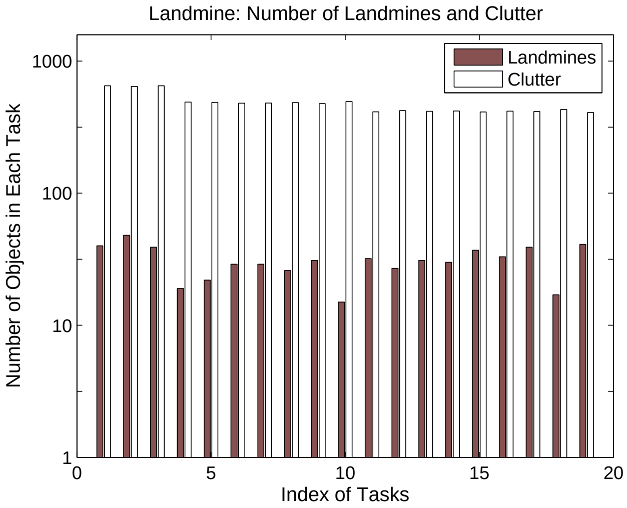


Figure 15: Number of landmines and clutter in each task for the landmine-detection data set (Xue et al., 2007).

structure, which is shared by all the tasks (Ando and Zhang, 2005). For the LR-MTL algorithm, we cite results on complete data from Xue et al. (2007), and implement the authors' Matlab code with default hyper-parameters on the cases with incomplete data. The Matlab implementation for the Structure-MTL algorithm is included in the "Transfer Learning Toolkit for Matlab" available at `http://multitask.cs.berkeley.edu/`. The dimension of the underlying structure is a user-set parameter, and it should be smaller than the feature dimension in the original space. As the dimension of the landmine detection data is 9, we set the hidden dimension as 5. We also tried 6, 7, and 8, and did not observe big differences in performance. Single-task learning (STL) iQGME and LR models are also included for comparison.

Each task is divided into training and test subsets randomly. Since the number of elements in the two classes is highly unbalanced, as shown in Figure 15, we impose that there is at least one instance from each class in each subset. Following Xue et al. (2007), the size of the training subset in each task varies from 20 to 300 in increments of 20, and 100 independent trials are performed for each size of data set. An average AUC (Hanley and McNeil, 1982) over all the 19 tasks is calculated as the performance representation for one trial of a given training size. Results are reported in Figure 16.

The first observation from Figure 16 is that we obtain a significant performance improvement for single-task learning by using the iQGME-VB instead of the linear logistic regression model (Xue et al., 2007). We also notice that the multi-task algorithm based on iQGME-VB further improves the performance when the training data are scarce, and yields comparable overall results as the LR-MTL does. The structure-MTL does not perform well on this data set. We suspect that a hidden structure in such a 9-dimensional space does not necessarily exist. Another possible reason may be that the minimization of empirical risk is sensitive for the cases with highly unbalanced labels, as for this data set.

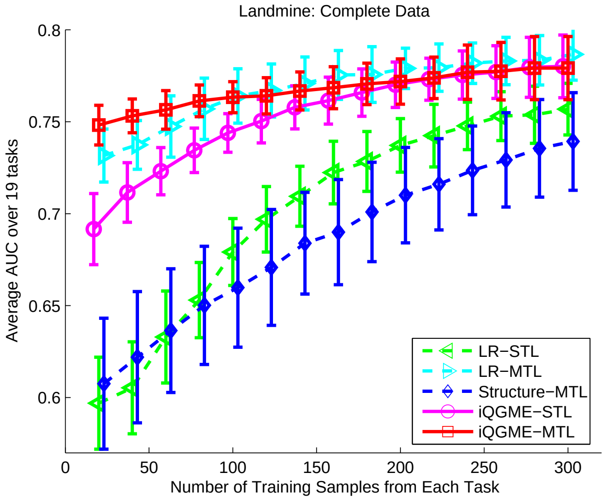
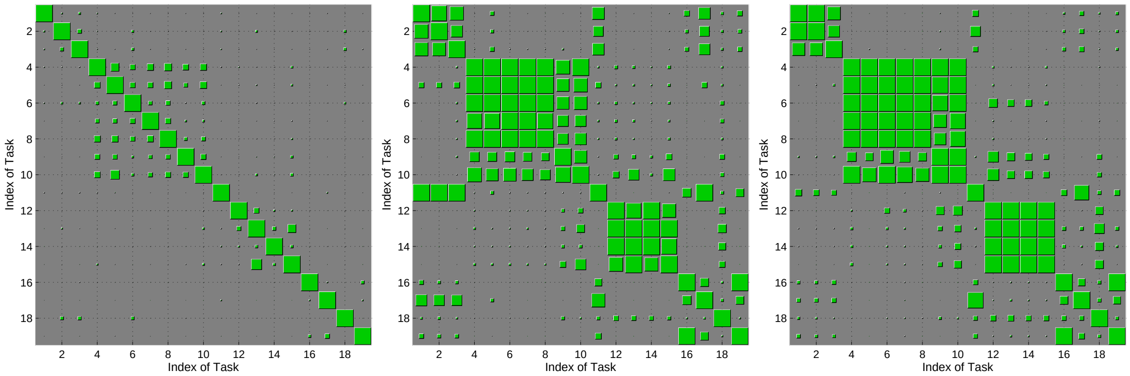


Figure 16: Average AUC over 19 tasks of landmine detection with complete data. Error bars reflect the standard deviation across 100 random partitions of training and test subsets. Results of logistic regression based algorithms are cited from Xue et al. (2007), where LR-MTL and LR-STL respectively correspond to SMTL-2 and STL in Figure 3 of Xue et al. (2007).

It is also interesting to explore the similarity among tasks. The similarity defined by different algorithms may be different. In Xue et al. (2007), two tasks are defined to be similar if they share the same linear classifier. However, with the joint distribution of covariates and the response, the iQGME-MTL requires both the data distributions and the classification boundaries to be similar if two tasks are deemed to be similar. Another difference is that two tasks could be partially similar since sharing between tasks is encouraged at the cluster-level instead of at the task-level (Xue et al. 2007 employs task-level clustering). We generate the similarity matrices between tasks as follows: In each random trial, there are in total S higher-level items shared among tasks. For each task, we can find the task-specific probability mass function (pmf) over all the higher-level items. Using these pmfs as the characteristics for tasks in the current trial, we calculate the pair-wise Kullback-Leibler (KL) distances and convert them to similarity measures through a minus exponential function. Results of multiple trials are summed over and normalized as shown in Figure 17. It can be seen that the similarity structure among tasks becomes clearer when we have more training data available. As discovered by Xue et al. (2007), we also find two big clusters correspond to two different vegetation conditions of the landmine fields (task 1-10 and task 11-19). Further sub-structures among tasks are also explored by the iQGME-MTL model, which may suggest other unknown difference among the landmine fields.

After yielding competitive results on the landmine-detection data set with complete data, the iQGME-based algorithms are evaluated on incomplete data, which are simulated by randomly removing a portion of feature values for each task as in Section 6.2. We consider three different



(a) (b) (c)

Figure 17: Similarity between tasks in the landmine detection problem with complete data given (a) 20, (b) 100, and (c) 300 training samples from each task. The size of green blocks represent the value of the corresponding matrix element.

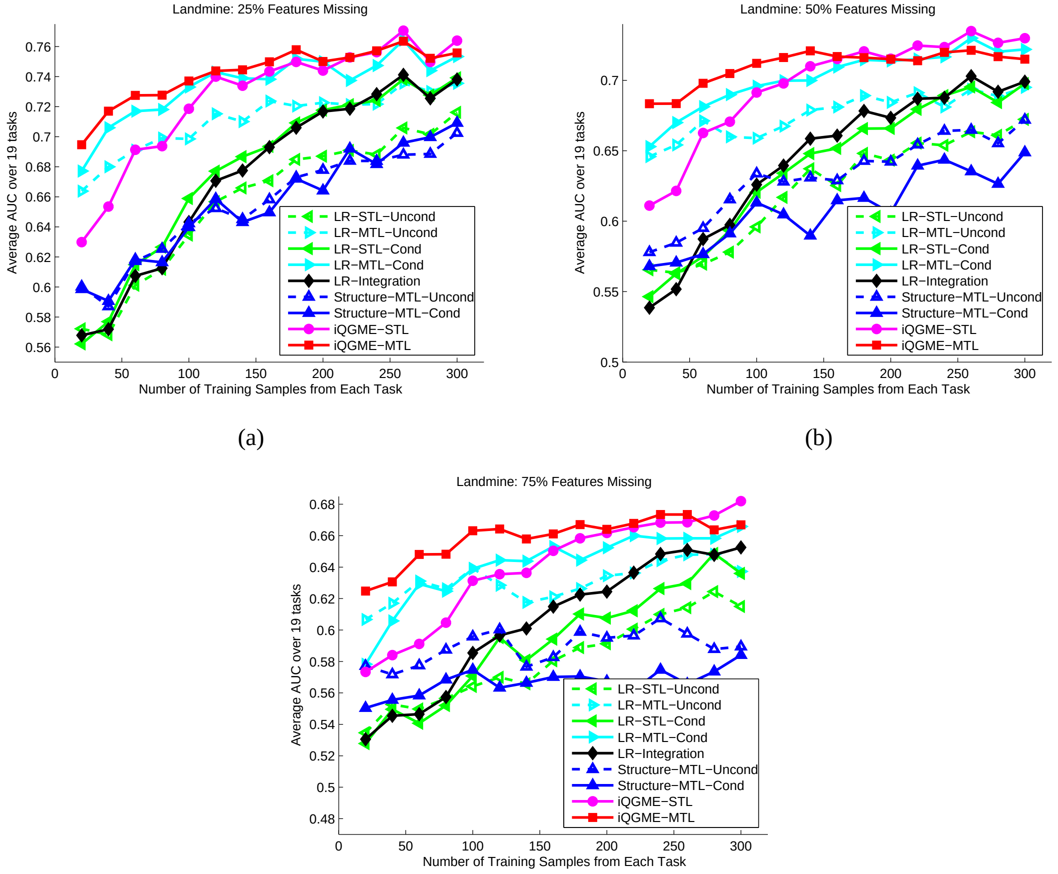
portions of missing values: 25%, 50% and 75%. As in the experiments above on benchmark data sets, we perform ten independent random trials for each setting of missing fraction and training size.

To the best of our knowledge, there exists no previous work in the literature on multi-task learning with missing data. As presented in Figure 18, we use the LR-MTL (Xue et al., 2007) and the Structure-MTL (Ando and Zhang, 2005) with missing values imputed as baseline algorithms. Results of the two-step LR with integration (Williams et al., 2007) and the LR-STL with single imputations are also included for comparison. Imputations using both unconditional-means and conditional-means are considered. From Figure 18, iQGME-STL consistently performs best among single-task learning methods and even better than LR-MTL-Uncond when the size of the training set is relatively large. The imputations using conditional-means yields consistently better results for the LR-based models on this data set. The iQGME-MTL outperforms the baselines and all the single-task learning methods overall. Furthermore, the improvement of iQGME-MTL is more pronounced when there are more features missing. These observations underscore the advantage of handling missing data in a principled manner and at the same time learning multiple tasks simultaneously.

The task-similarity matrices for the incomplete-data cases are shown in Figure 19. It can be seen that when a small fraction (e.g., 25%) of the feature values are missing and training data are rich (e.g., 300 samples from each task), the similarity pattern among tasks is similar to what we have seen for the complete-data case. As the fraction of missing values becomes larger, tasks appear more different from each other in terms of the usage of the higher-level items. Considering that the missing pattern for each task is unique, it is probable that tasks look quite different from each other after a large fraction of feature values are missing. However, the fact that tasks tend to use different subsets of higher-level items does not mean it is equivalent to learning them separately (STL), as parameters of the common base measures are inferred based on all the tasks.

6.6 Multi-Task Learning with Handwritten Letters Data

The final example corresponds to multi-task learning of classifiers for handwritten letters, this data set included in the "Transfer Learning Toolkit for Matlab" available at `http://multitask.cs.`



(a)

(b)

(c)

Figure 18: Average AUC over 19 tasks of landmine detection for the cases when (a) 25%, (b) 50%, and (c) 75% of the features are missing. Mean values of performance across 10 random partitions of training and test subsets are reported. Error bars are omitted for legibility.

`berkeley.edu/`. The objective of each task is to distinguish two letters which are easily confused. The number of samples for all the letters considered in the total eight tasks is summarized in Table 2. Each sample is a 16×8 image as shown in Figure 20. We use the 128 pixel values of each sample directly as its feature vector.

Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8
‘c’: 2107	‘g’: 2460	‘m’: 1596	‘a’: 4016	‘i’: 4895	‘a’: 4016	‘f’: 918	‘h’: 858
‘e’: 4928	‘y’: 1218	‘n’: 5004	‘g’: 2460	‘j’: 188	‘o’: 3880	‘t’: 2131	‘n’: 5004

Table 2: Handwritten letters classification data set.

We compare the proposed iQGME-MTL algorithm to the LR-MTL (Xue et al., 2007) and the Structure-MTL (Ando and Zhang, 2005) mentioned in Section 6.5. For the non-parametric Bayesian methods (iQGME-MTL and LR-MTL), we use the same parameter setting as before. The dimension

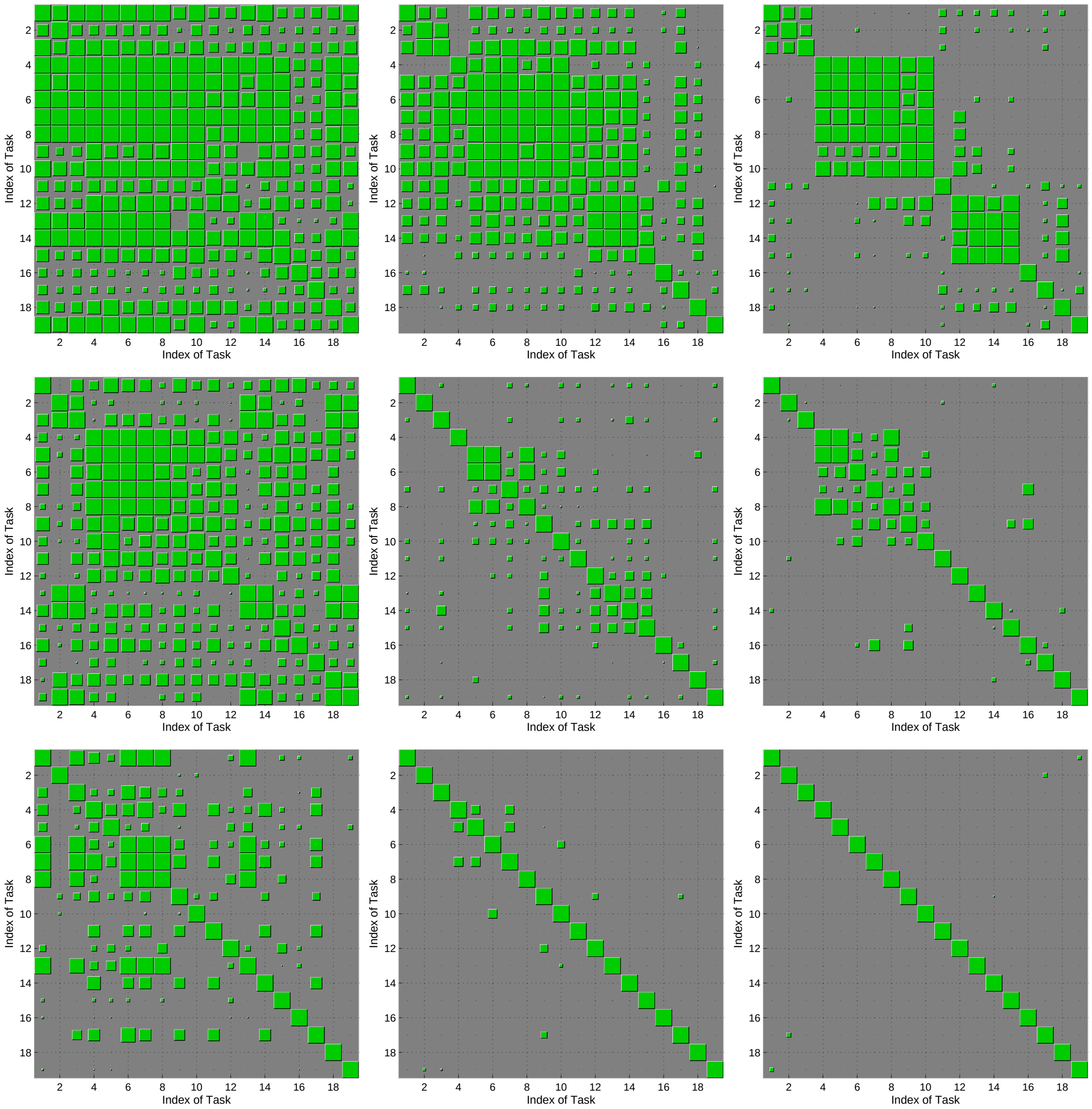


Figure 19: Similarity between tasks in the landmine detection problem with incomplete data. Row 1, 2 and 3 corresponds to the cases with 25%, 50% and 75% features missing, respectively; column 1, 2 and 3 corresponds to the cases with 20, 100 and 300 training samples from each task, respectively.

of the underlying structure for the Structure-MTL is set to be 50 in the results shown in Figure 21. We also tried 10, 20, 40, 60, 80 and 100, and did not observe big difference. From Figure 21, the iQGME-MTL performs significantly better than the baselines on this data set for all the missing fractions and training fractions under consideration. As we expected, the Structure-MTL yields comparable results as the LR-MTL on this data set.

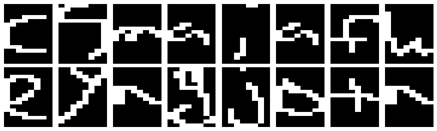
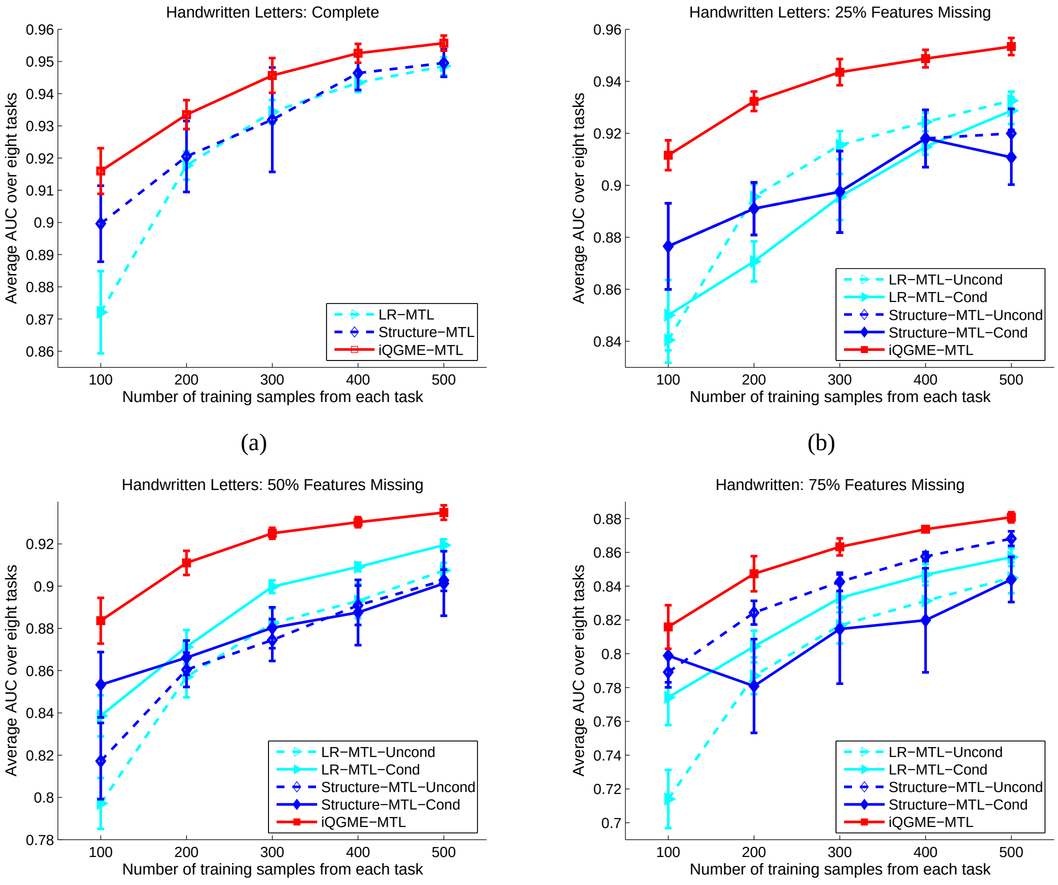


Figure 20: Sample images of the handwritten letters. The two images in each column represents the two classes in the corresponding task described in Table 2.



(a)

(b)

(c)

(d)

Figure 21: Average AUC over eight tasks of handwriting letters classification for the cases when (a) none, (b) 25%, (c) 50%, and (d) 75% of the features are missing. Mean values of performance with one standard deviation across 10 random partitions of training and test subsets are reported.

7. Conclusion and Future Work

In this paper we have introduced three new concepts, summarized as follows. First, we have employed non-parametric Bayesian techniques to develop a mixture-of-experts algorithm for classifier design, which employs a set of localized (in feature space) linear classifiers as experts. The Dirichlet process is employed to allow the model to infer automatically the proper number of experts and their characteristics; in fact, since a Bayesian formulation is employed, a full posterior distribution is manifested on the properties of the local experts, including their number. Secondly, the classifier is endowed with the ability to naturally address missing data, without the need for an imputation step. Finally, the whole framework has been placed within the context of a multi-task learning, allowing one to jointly infer classifiers for multiple data sets with missing data. The multi-task-learning component has also been implemented with the general tools associated with the Dirichlet process, with specific implementations manifested via the hierarchical Dirichlet process. Because of the hierarchical form of the model, in terms of a sequence of distributions in the conjugate-exponential family, all inference has been manifested efficiently via variational Bayesian (VB) analysis. The VB results have been compared to those computed via Gibbs sampling; the VB results have been found to be consistent with those inferred via Gibbs sampling, while requiring a small fraction of the computational costs. Results have been presented for single-task and multi-task learning on various data sets with the same hyper-parameters setting (no model-parameter tuning), and encouraging algorithm performance has been demonstrated.

Concerning future research, we note that the use of multi-task learning provides an important class of contextual information, and therefore is particularly useful when one has limited labeled data and when the data are incomplete (missing features). Another form of context that has received significant recent attention is semi-supervised learning (Zhu, 2005). There has been recent work on integrating multi-task learning with semi-supervised learning (Liu et al., 2007). An important new research direction includes extending semi-supervised multi-task learning to realistic problems for which the data are incomplete.

Appendix A.

The update equations of single-task learning with incomplete data are summarized as follows:

1. $q(t_i|\mu_i^t)$

$$\mu_i^t = \sum_{h=1}^{N} \rho_{ih} \langle \boldsymbol{w}_h \rangle^T \hat{\boldsymbol{x}^b}_{i,h} \quad \text{where} \quad \hat{\boldsymbol{x}^b}_{i,h} = [\boldsymbol{x}_i^{o_i}; \boldsymbol{m}_h^{m_i|o_i}; 1].$$

The expectation of t_i and t_i^2 may be derived according to properties of truncated normal distributions:

$$\begin{aligned} \langle t_i \rangle &= \mu_i^t + \frac{\phi(-\mu_i^t)}{\mathbf{1}(y_i = 1) - \Phi(-\mu_i^t)}, \\ \langle t_i^2 \rangle &= 1 + (\mu_i^t)^2 + \frac{\mu_i^t \phi(-\mu_i^t)}{\mathbf{1}(y_i = 1) - \Phi(-\mu_i^t)}, \end{aligned}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ denote the probability density function and the cumulative density function of the standard normal distribution, respectively.

2. $q(\boldsymbol{x}_i^{m_i}, z_i | \boldsymbol{m}_{ih}^{m_i|o_i}, \boldsymbol{\Sigma}_{ih}^{m_i|o_i}, \rho_{ih})$
A related derivation for a GMM model with incomplete data could be found in Williams et al. (2007), where no classifier terms appear.

First, we explicitly write the intercept w_h^b , that is, $\boldsymbol{w}_h = [(\boldsymbol{w}_h^x)^T, w_h^b]^T$:

$$\begin{aligned}
& q(\boldsymbol{x}_i^{m_i}, z_i = h) \\
\propto \; & \exp\{\langle \ln[p(t_i|z_i = h, \boldsymbol{x}_i, \boldsymbol{W}) p(z_i = h|V) p(\boldsymbol{x}_i|z_i = h, \boldsymbol{\mu}, \boldsymbol{\Lambda})]\rangle_{q(t_i)q(\boldsymbol{w}_h)q(V)q(\boldsymbol{\mu}_h,\boldsymbol{\Lambda}_h)}\} \\
\propto \; & A_{ih}\mathcal{N}_P(\boldsymbol{x}_i|\tilde{\boldsymbol{\mu}}_{ih}, \tilde{\boldsymbol{\Sigma}}_{ih}),
\end{aligned}$$

where

$$\begin{aligned}
\tilde{\boldsymbol{\Sigma}}_{ih} &= [\langle \boldsymbol{w}_h^x(\boldsymbol{w}_h^x)^T\rangle + \nu_h \boldsymbol{B}_h]^{-1} \\
\tilde{\boldsymbol{\mu}}_{ih} &= \tilde{\boldsymbol{\Sigma}}_{ih}[\langle t_i\rangle\langle \boldsymbol{w}_h^x\rangle + \nu_h \boldsymbol{B}_h \boldsymbol{m}_h - \langle \boldsymbol{w}_h^x w_h^b\rangle] \\
A_{ih} &= \exp\{\langle \ln V_h\rangle + \sum_{l<h}\langle \ln(1-V_l)\rangle + \langle t_i\rangle\langle w_h^b\rangle \\
&\quad + \frac{1}{2}[\langle \ln|\boldsymbol{\Lambda}_h|\rangle + \tilde{\boldsymbol{\mu}}_{ih}^T \tilde{\boldsymbol{\Sigma}}_{ih}^{-1}\tilde{\boldsymbol{\mu}}_{ih} + \ln|\tilde{\boldsymbol{\Sigma}}_{ih}| - \frac{P}{u_h} - \boldsymbol{m}_h^T \nu_h \boldsymbol{B}_h \boldsymbol{m}_h - \langle (w_h^b)^2\rangle]\}.
\end{aligned}$$

Since

$$\begin{bmatrix} \boldsymbol{x}_i^{o_i} \\ \boldsymbol{x}_i^{m_i} \end{bmatrix} \sim \mathcal{N}_P\left(\begin{bmatrix} \tilde{\boldsymbol{\mu}}_{ih}^{o_i} \\ \tilde{\boldsymbol{\mu}}_{ih}^{m_i} \end{bmatrix}, \begin{bmatrix} \tilde{\boldsymbol{\Sigma}}_{ih}^{o_io_i} & \tilde{\boldsymbol{\Sigma}}_{ih}^{o_im_i} \\ \tilde{\boldsymbol{\Sigma}}_{ih}^{m_io_i} & \tilde{\boldsymbol{\Sigma}}_{ih}^{m_im_i} \end{bmatrix}\right),$$

the conditional distribution of missing features $\boldsymbol{x}_i^{m_i}$ given observable features $\boldsymbol{x}_i^{o_i}$ is also a normal distribution, that is, $\boldsymbol{x}_i^{m_i}|\boldsymbol{x}_i^{o_i} \sim \mathcal{N}_{|m_i|}(\boldsymbol{m}_h^{m_i|o_i}, \boldsymbol{\Sigma}_h^{m_i|o_i})$ with

$$\begin{aligned}
\boldsymbol{m}_h^{m_i|o_i} &= \tilde{\boldsymbol{\mu}}_{ih}^{m_i} + \tilde{\boldsymbol{\Sigma}}_{ih}^{m_io_i}(\tilde{\boldsymbol{\Sigma}}_{ih}^{o_io_i})^{-1}(\boldsymbol{x}_i^{o_i} - \tilde{\boldsymbol{\mu}}_{ih}^{o_i}), \\
\boldsymbol{\Sigma}_h^{m_i|o_i} &= \tilde{\boldsymbol{\Sigma}}_{ih}^{m_im_i} - \tilde{\boldsymbol{\Sigma}}_{ih}^{m_io_i}(\tilde{\boldsymbol{\Sigma}}_{ih}^{o_io_i})^{-1}\tilde{\boldsymbol{\Sigma}}_{ih}^{o_im_i}.
\end{aligned}$$

Therefore, $q(\boldsymbol{x}_i^{m_i}, z_i = h)$ could be factorized as the product of a factor independent of $\boldsymbol{x}_i^{m_i}$ and the variational posterior of $\boldsymbol{x}_i^{m_i}$, that is,

$$\begin{aligned}
q(\boldsymbol{x}_i^{m_i}, z_i = h) &\propto A_{ih}\mathcal{N}_{|o_i|}(\boldsymbol{x}_i^{o_i}|\tilde{\boldsymbol{\mu}}_{ih}^{o_i}, \tilde{\boldsymbol{\Sigma}}_{ih}^{o_io_i})\mathcal{N}_{|m_i|}(\boldsymbol{x}_i^{m_i}|\boldsymbol{m}_h^{m_i|o_i}, \boldsymbol{\Sigma}_h^{m_i|o_i}) \\
\rho_{ih} &\propto A_{ih}\mathcal{N}_{|o_i|}(\boldsymbol{x}_i^{o_i}|\tilde{\boldsymbol{\mu}}_{ih}^{o_i}, \tilde{\boldsymbol{\Sigma}}_{ih}^{o_io_i})
\end{aligned}$$

For complete data, no factorization for the distribution for $\boldsymbol{x}_i^{m_i}$ is necessary:

$$\begin{aligned}
\rho_{ih} \propto \; & \exp\{\langle t_i\rangle\langle \boldsymbol{w}_h\rangle^T \boldsymbol{x}_i - \frac{1}{2}\boldsymbol{x}_i^T\langle \boldsymbol{w}_h\boldsymbol{w}_h^T\rangle \boldsymbol{x}_i + \langle \ln V_h\rangle \\
& + \sum_{l<h}\langle \ln(1-V_l)\rangle + \frac{1}{2}\langle \ln|\boldsymbol{\Lambda}_h|\rangle - \frac{1}{2}\langle (\boldsymbol{x}_i - \boldsymbol{\mu}_h)^T\boldsymbol{\Lambda}_h(\boldsymbol{x}_i - \boldsymbol{\mu}_h)\rangle\}
\end{aligned}$$

3. $q(V_h|v_{h1}, v_{h2})$

   Similar updating could be found in Blei and Jordan (2006), except that we put a prior belief on α here instead of setting a fixed number.

$$\begin{aligned}
& v_{h1} = 1 + \sum_{i=1}^{n}\rho_{ih}, \quad v_{h2} = \langle \alpha\rangle + \sum_{i=1}^{n}\sum_{l>h}\rho_{il}; \\
& \langle \ln V_h\rangle = \psi(v_{h1}) - \psi(v_{h1} + v_{h2}), \quad \langle \ln(1-V_l)\rangle = \psi(v_{l2}) - \psi(v_{l1} + v_{l2}).
\end{aligned}$$

4. $q(\boldsymbol{\mu}_h, \boldsymbol{\Lambda}_h | \boldsymbol{m}_h, u_h, \boldsymbol{B}_h, \nu_h)$

   Similar updating could be found in Williams et al. (2007).

$$\nu_h = \nu_0 + N_h, \quad u_h = u_0 + N_h, \quad \boldsymbol{m}_h = \frac{u_0 \boldsymbol{m}_0 + N_h \bar{\boldsymbol{x}}_h}{u_h},$$

$$\boldsymbol{B}_h^{-1} = \boldsymbol{B}_0^{-1} + \sum_{i=1}^{n} \rho_{ih} \hat{\boldsymbol{\Omega}}_{i,h} + N_h \bar{\boldsymbol{S}}_h + \frac{u_0 N_h}{u_h} (\bar{\boldsymbol{x}}_h - \boldsymbol{m}_0)(\bar{\boldsymbol{x}}_h - \boldsymbol{m}_0)^T,$$

   where

$$N_h = \sum_{i=1}^{n} \rho_{ih}, \quad \bar{\boldsymbol{x}}_h = \sum_{i=1}^{n} \rho_{ih} \boldsymbol{x}_i / N_h, \quad \bar{\boldsymbol{S}}_h = \sum_{i=1}^{n} \rho_{ih} (\hat{\boldsymbol{x}}_{i,h} - \bar{\boldsymbol{x}}_h)(\hat{\boldsymbol{x}}_{i,h} - \bar{\boldsymbol{x}}_h)^T / N_h.$$

$$\hat{\boldsymbol{x}}_{i,h} = \begin{bmatrix} \boldsymbol{x}_i^{o_i} \\ \boldsymbol{m}_h^{m_i|o_i} \end{bmatrix}, \quad \hat{\boldsymbol{\Omega}}_{i,h} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_h^{m_i|o_i} \end{bmatrix}.$$

$$\langle \ln |\boldsymbol{\Lambda}_h| \rangle = \sum_{p=1}^{P} \psi((\nu_h - p + 1)/2) + P \ln 2 + \ln |\boldsymbol{B}_h|,$$

$$\langle (\boldsymbol{x}_i - \boldsymbol{\mu}_h)^T \boldsymbol{\Lambda}_h (\boldsymbol{x}_i - \boldsymbol{\mu}_h) \rangle = (\hat{\boldsymbol{x}}_{i,h} - \boldsymbol{m}_h)^T \nu_h \boldsymbol{B}_h (\hat{\boldsymbol{x}}_{i,h} - \boldsymbol{m}_h) + P/u_h + \mathrm{tr}(\nu_h \boldsymbol{B}_h \hat{\boldsymbol{\Omega}}_{i,h}).$$

5. $q(\boldsymbol{w}_h | \boldsymbol{\mu}_h^w, \boldsymbol{\Sigma}_h^w)$, $\langle \boldsymbol{w}_h \rangle = \boldsymbol{\mu}_h^w$, $\langle \boldsymbol{w}_h \boldsymbol{w}_h^T \rangle = \boldsymbol{\Sigma}_h^w + \boldsymbol{\mu}_h^w (\boldsymbol{\mu}_h^w)^T$.

$$\boldsymbol{\Sigma}_h^w = \left(\sum_{i=1}^{n} \rho_{ih} (\hat{\boldsymbol{\Omega}}_{i,h} + \hat{\boldsymbol{x}^b}_{i,h} \hat{\boldsymbol{x}^b}_{i,h}^T) + \mathrm{diag}(\langle \boldsymbol{\lambda} \rangle) \right)^{-1},$$

$$\boldsymbol{\mu}_h^w = \boldsymbol{\Sigma}_h^w \left(\sum_{i=1}^{n} \rho_{ih} \hat{\boldsymbol{x}^b}_{i,h} \langle t_i \rangle + \mathrm{diag}(\langle \boldsymbol{\lambda} \rangle) \boldsymbol{\phi} \right).$$

6. $q(\zeta_p, \lambda_p | \phi_p, \gamma, a_p, b_p)$, $\langle \lambda_p \rangle = a_p / b_p$.

   Similar updating could be found in Xue et al. (2007).

$$\phi_p = \sum_{h=1}^{N} \langle w_{hp} \rangle / \gamma, \quad \gamma = \gamma_0 + N$$

$$a_p = a_0 + \frac{N}{2}, \quad b_p = b_0 + \frac{1}{2} \sum_{h=1}^{N} \langle w_{hp}^2 \rangle - \frac{1}{2} \gamma \phi_p^2.$$

7. $q(\alpha | \tau_1, \tau_2)$, $\langle \alpha \rangle = \tau_1 / \tau_2$.

   Similar updating could be found in any VB-inferred DP model with a Gamma prior on α (Xue et al., 2007).

$$\tau_1 = N - 1 + \tau_{10}, \quad \tau_2 = \tau_{20} - \sum_{h=1}^{N-1} \langle \ln(1 - V_h) \rangle.$$

The update equations of multi-task learning with incomplete data are summarized as follows:

1. $q(t_{ji}|\mu^t_{ji})$

$$\mu^t_{ji} = \sum_{s=1}^{S} E\sigma_{jis}\langle \boldsymbol{w}_s\rangle^T \hat{\boldsymbol{x}^b}_{ji} \quad \text{where} \quad E\sigma_{jis} = \sum_{h=1}^{N} \rho_{jih}\sigma_{jhs}, \quad \hat{\boldsymbol{x}^b}_{ji} = [\boldsymbol{x}^{o_{ji}}_{ji}; \boldsymbol{m}^{m_{ji}|o_{ji}}_{ji}; 1].$$

2. $q(x^{m_{ji}}_{ji}|\boldsymbol{m}^{m_{ji}|o_{ji}}_{ji}, \boldsymbol{\Sigma}^{m_{ji}|o_{ji}}_{ji})$

$$\begin{aligned}\boldsymbol{m}^{m_{ji}|o_{ji}}_{ji} &= \tilde{\boldsymbol{\mu}}^{m_{ji}}_{ji} + \tilde{\boldsymbol{\Sigma}}^{m_{ji}o_{ji}}_{ji}(\tilde{\boldsymbol{\Sigma}}^{o_{ji}o_{ji}}_{ji})^{-1}(\boldsymbol{x}^{o_{ji}}_{ji} - \tilde{\boldsymbol{\mu}}^{o_{ji}}_{ji}),\\ \boldsymbol{\Sigma}^{m_{ji}|o_{ji}}_{ji} &= \tilde{\boldsymbol{\Sigma}}^{m_{ji}m_{ji}}_{ji} - \tilde{\boldsymbol{\Sigma}}^{m_{ji}o_{ji}}_{ji}(\tilde{\boldsymbol{\Sigma}}^{o_{ji}o_{ji}}_{ji})^{-1}\tilde{\boldsymbol{\Sigma}}^{o_{ji}m_{ji}}_{ji},\end{aligned}$$

where

$$\begin{aligned}\tilde{\boldsymbol{\Sigma}}_{ji} &= \left(\sum_{s=1}^{S} E\sigma_{jis}(\langle \boldsymbol{w}^x_s(\boldsymbol{w}^x_s)^T\rangle + \nu_s\boldsymbol{B}_s)\right)^{-1},\\ \tilde{\boldsymbol{\mu}}_{ji} &= \tilde{\boldsymbol{\Sigma}}_{ji}\sum_{s=1}^{S} E\sigma_{jis}(\langle t_{ji}\rangle\langle \boldsymbol{w}^x_s\rangle + \nu_s\boldsymbol{B}_s\boldsymbol{m}_s - \langle \boldsymbol{w}^x_s w^b_s\rangle).\end{aligned}$$

$$\hat{\boldsymbol{x}}_{ji} = \begin{bmatrix} \boldsymbol{x}^{o_{ji}}_{ji} \\ \boldsymbol{m}^{m_{ji}|o_{ji}}_{ji} \end{bmatrix}, \quad \hat{\boldsymbol{\Omega}}_{ji} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}^{m_{ji}|o_{ji}}_{ji} \end{bmatrix}, \quad \langle \boldsymbol{x}_{ji}\boldsymbol{x}^T_{ji}\rangle = \hat{\boldsymbol{x}}_{ji}\hat{\boldsymbol{x}}^T_{ji} + \hat{\boldsymbol{\Omega}}_{ji}.$$

3. $q(z_{ji}|\boldsymbol{\rho}_{ji})$

$$\begin{aligned} & \rho_{jih} = q(z_{ji} = h)\\ \propto\ & \exp\{\sum_{s=1}^{S}\sigma_{jhs}[\langle t_{ji}\rangle\langle \boldsymbol{w}_s\rangle^T\hat{\boldsymbol{x}^b}_{ji} - \frac{1}{2}\mathrm{tr}(\langle \boldsymbol{w}_s\boldsymbol{w}^T_s\rangle\langle \boldsymbol{x}^b_{ji}(\boldsymbol{x}^b_{ji})^T\rangle)]\\ & +\langle \ln V_{jh}\rangle + \sum_{l<h}\langle \ln(1-V_{jl})\rangle\\ & +\frac{1}{2}\sum_{s=1}^{S}\sigma_{jhs}[\langle \ln|\boldsymbol{\Lambda}_s|\rangle - (\hat{\boldsymbol{x}}_{ji} - \boldsymbol{m}_s)^T\nu_s\boldsymbol{B}_s(\hat{\boldsymbol{x}}_{ji} - \boldsymbol{m}_s) - P/u_s - \mathrm{tr}(\nu_s\boldsymbol{B}_s\hat{\boldsymbol{\Omega}}_{ji})]\}.\end{aligned}$$

4. $q(\boldsymbol{V}|\boldsymbol{v})$

$$v_{jh1} = 1 + \sum_{i=1}^{n_j}\rho_{jih}, \qquad v_{jh2} = \langle \alpha\rangle + \sum_{l>h}\sum_{i=1}^{n_j}\rho_{jil}.$$

$$\langle \ln V_{jh}\rangle = \psi(v_{jh1}) - \psi(v_{jh1} + v_{jh2}), \qquad \langle \ln(1-V_{jl})\rangle = \psi(v_{jl2}) - \psi(v_{jl1} + v_{jl2}).$$

5. $q(\alpha|\tau_1,\tau_2)$, $\langle \alpha\rangle = \tau_1/\tau_2$.

$$\tau_1 = J(N-1) + \tau_{10}, \quad \tau_2 = \tau_{20} - \sum_{j=1}^{J}\sum_{h=1}^{N-1}\langle \ln(1-V_{jh})\rangle.$$

6. $q(\boldsymbol{c}|\boldsymbol{\sigma})$

$$\begin{aligned}
\sigma_{jhs} \quad \propto \quad & \exp\{\sum_{i=1}^{n_j} \rho_{jih}[\langle t_{ji}\rangle \langle \boldsymbol{w}_s\rangle^T \hat{\boldsymbol{x}^b}_{ji} - \frac{1}{2}\mathrm{tr}(\langle \boldsymbol{w}_s \boldsymbol{w}_s^T\rangle \langle \boldsymbol{x}^b_{ji}(\boldsymbol{x}^b_{ji})^T\rangle)] \\
& + \langle \ln U_s\rangle + \sum_{l<s}\langle \ln(1-U_s)\rangle \\
& + \frac{1}{2}\sum_{i=1}^{n_j} \rho_{jih}[\langle \ln|\boldsymbol{\Lambda}_s|\rangle - (\hat{\boldsymbol{x}}_{ji} - \boldsymbol{m}_s)^T \nu_s \boldsymbol{B}_s(\hat{\boldsymbol{x}}_{ji} - \boldsymbol{m}_s) - P/u_s - \mathrm{tr}(\nu_s \boldsymbol{B}_s \hat{\boldsymbol{\Omega}}_{ji})]\}.
\end{aligned}$$

7. $q(U_s|\kappa_{s1}, \kappa_{s2})$

$$\kappa_{s1} = 1 + \sum_{j=1}^{J}\sum_{h=1}^{N}\sigma_{jhs}, \quad \kappa_{s2} = \langle\beta\rangle + \sum_{j=1}^{J}\sum_{h=1}^{N}\sum_{l>s}\sigma_{jhl}.$$

$$\langle \ln U_s\rangle = \psi(\kappa_{s1}) - \psi(\kappa_{s1}+\kappa_{s2}), \quad \langle \ln(1-U_s)\rangle = \psi(\kappa_{s2}) - \psi(\kappa_{s1}+\kappa_{s2}).$$

8. $q(\beta|\tau_3, \tau_4)$, $\langle\beta\rangle = \tau_3/\tau_4$.

$$\tau_3 = S - 1 + \tau_{30}, \quad \tau_4 = \tau_{40} - \sum_{s=1}^{S-1}\langle \ln(1-U_s)\rangle.$$

9. $q(\boldsymbol{\mu}_s, \boldsymbol{\Lambda}_s|\boldsymbol{m}_s, u_s, \boldsymbol{B}_s, \nu_s)$

$$\nu_s = \nu_0 + N_s, \quad u_s = u_0 + N_s, \quad \boldsymbol{m}_s = \frac{u_0\boldsymbol{m}_0 + N_s\bar{\boldsymbol{x}}_s}{u_s},$$

$$\boldsymbol{B}_s^{-1} = \boldsymbol{B}_0^{-1} + \sum_{j=1}^{J}\sum_{i=1}^{n_j} E\sigma_{jis}\hat{\boldsymbol{\Omega}}_{ji} + N_s\bar{\boldsymbol{S}}_s + \frac{u_0 N_s}{u_s}(\bar{\boldsymbol{x}}_s - \boldsymbol{m}_0)(\bar{\boldsymbol{x}}_s - \boldsymbol{m}_0)^T,$$

where $E\sigma_{jis} = \sum_{h=1}^{N}\rho_{jih}\sigma_{jhs}$, and

$$N_s = \sum_{j=1}^{J}\sum_{i=1}^{n_j} E\sigma_{jis}, \quad \bar{\boldsymbol{x}}_s = \sum_{j=1}^{J}\sum_{i=1}^{n_j} E\sigma_{jis}\hat{\boldsymbol{x}}_{ji}/N_s,$$

$$\bar{\boldsymbol{S}}_s = \sum_{j=1}^{J}\sum_{i=1}^{n_j} E\sigma_{jis}(\hat{\boldsymbol{x}}_{ji} - \bar{\boldsymbol{x}}_s)(\hat{\boldsymbol{x}}_{ji} - \bar{\boldsymbol{x}}_s)^T/N_s.$$

$$\langle \ln|\boldsymbol{\Lambda}_s|\rangle = \sum_{p=1}^{P}\psi((\nu_s - p + 1)/2) + P\ln 2 + \ln|\boldsymbol{B}_s|,$$

$$\langle(\boldsymbol{x}_{ji} - \boldsymbol{\mu}_s)^T\boldsymbol{\Lambda}_s(\boldsymbol{x}_{ji} - \boldsymbol{\mu}_s)\rangle = (\hat{\boldsymbol{x}}_{ji} - \boldsymbol{m}_s)^T\nu_s\boldsymbol{B}_s(\hat{\boldsymbol{x}}_{ji} - \boldsymbol{m}_s) + P/u_s + \mathrm{tr}(\nu_s\boldsymbol{B}_s\hat{\boldsymbol{\Omega}}_{ji}).$$

10. $q(\boldsymbol{w}_s|\boldsymbol{\mu}_s^w, \boldsymbol{\Sigma}_s^w)$

$$\begin{aligned}
\boldsymbol{\Sigma}_s^w \quad &= \quad \left(\sum_{j=1}^{J}\sum_{i=1}^{n_j} E\sigma_{jis}(\hat{\boldsymbol{x}^b}_{ji}\hat{\boldsymbol{x}^b}_{ji}^T + \hat{\boldsymbol{\Omega}}_{ji}) + \mathrm{diag}(\langle\boldsymbol{\lambda}\rangle)\right)^{-1}, \\
\boldsymbol{\mu}_s^w \quad &= \quad \boldsymbol{\Sigma}_s^w\left(\sum_{j=1}^{J}\sum_{i=1}^{n_j} E\sigma_{jis}\hat{\boldsymbol{x}^b}_{ji}\langle t_{ji}\rangle + \mathrm{diag}(\langle\boldsymbol{\lambda}\rangle)\phi\right).
\end{aligned}$$

$$\langle\boldsymbol{w}_s\rangle = \boldsymbol{\mu}_s^w, \qquad \langle\boldsymbol{w}_s\boldsymbol{w}_s^T\rangle = \boldsymbol{\Sigma}_s^w + \boldsymbol{\mu}_s^w(\boldsymbol{\mu}_s^w)^T.$$

11. $q(\lambda_p|a_p,b_p)$, $\langle\lambda_p\rangle = a_p/b_p$.

$$\phi_p = \sum_{s=1}^{S}\langle w_{sp}\rangle/\gamma, \quad \gamma = \gamma_0 + S,$$
$$a_p = a_0 + \frac{S}{2}, \quad b_p = b_0 + \frac{1}{2}\sum_{s=1}^{S}\langle W_{sp}^2\rangle - \frac{1}{2}\gamma\phi_p^2.$$

References

J. H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88:669–679, 1993.

R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

H. Attias. A variational Bayesian framework for graphical models. In *Advances in Neural Information Processing Systems (NIPS)*, 2000.

M. J. Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD dissertation, University College London, Gatsby Computational Neuroscience Unit, 2003.

D. M. Blei and M. I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–144, 2006.

R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

G. Chechik, G. Heitz, G. Elidan, P. Abbeel, and D. Koller. Max-margin classification of data with absent features. *Journal of Machine Learning Research*, 9:1–21, 2008.

L. M. Collins, J. L. Schafer, and C. M. Kam. A comparison of inclusive and restrictive strategies in modern missing data procedures. *Psychological Methods*, 6(4):330–351, 2001.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society B*, 39:1–38, 1977.

U. Dick, P. Haider, and T. Scheffer. Learning from incomplete data with infinite imputations. In *International Conference on Machine Learning (ICML)*, 2008.

D. B. Dunson, N. Pillai, and J.-H. Park. Bayesian density regression. *Journal of the Royal Statistical Society: Series B*, 69, 2007.

M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90:577–588, 1995.

T. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1: 209–230, 1973.

A. E. Gelfand, S. E. Hills, A. Racine-Poon, and A. F. M. Smith. Illustration of Bayesian inference in normal data models using Gibbs sampling. *Journal of American Statistical Association*, 85: 972–985, 1990.

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

Z. Ghahramani and G. E. Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, Department of Computer Science, University of Toronto, 1996.

Z. Ghahramani and M. I. Jordan. Learning from incomplete data. Technical report, Massachusetts Institute of Technology, 1994.

T. Graepel. Kernel matrix completion by semidefinite programming. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 694–699, 2002.

J. Hanley and B. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143:29–36, 1982.

L. Hannah, D. Blei, and W. Powell. Dirichlet process mixtures of generalized linear models. In *Artificial Intelligence and Statistics (AISTATS)*, pages 313–320, 2010.

J. Ibrahim. Incomplete data in generalized linear models. *Journal of the American Statistical Association*, 85:765–769, 1990.

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

R. A. Jacobs, M. I. Jordon, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.

M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.

K. Kurihara, M. Welling, and Y. W. Teh. Collapsed variational Dirichlet process mixture models. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2796–2801, 2007.

Percy Liang and Michael I. Jordan. An asymptotic analysis of generative, discriminative, and pseudolikelihood estimators. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 584–591, 2008.

X. Liao, H. Li, and L. Carin. Quadratically gated mixture of experts for incomplete data classification. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 553–560, 2007.

Q. Liu, X. Liao, and L. Carin. Semi-supervised multitask learning. In *Neural Information Processing Systems*, 2007.

S. N. MacEachern and P. Müller. Estimating mixture of Dirichlet process models. *Journal of Computational and Graphical Statistics*, 7, 1998.

E. Meeds and S. Osindero. An alternative infinite mixture of Gaussian process experts. In *NIPS 18*, pages 883–890. MIT Press, 2006.

P. Müller, A. Erkanli, and M. West. Bayesian curve fitting using multivariate normal mixtures. *Biometrika*, 83:67–79, 1996.

R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical report, Department of Computer Science, University of Toronto, 1993.

D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases. *http://www.ics.uci.edu/∼mlearn/MLRepository.html*, 1998.

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

C. E. Rasmussen and Z. Ghahramani. Infinite mixtures of Gaussian process experts. In *NIPS 14*. MIT Press, 2002.

A. Rodríguez, D. B. Dunson, and A. E. Gelfang. Bayesian nonparametric functional data analysis through density estimation. *Biometrika*, 96, 2009.

D. B. Rubin. Inference and missing data. *Biometrika*, 63:581–592, 1976.

D. B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. John Wiley & Sons, Inc., 1987.

J. L. Schafer and J. W. Graham. Missing data: Our view of the state of the art. *Psychological Methods*, 7:147–177, 2002.

J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 1:639–650, 1994.

B. Shahbaba and R. Neal. Nonlinear models using Dirichlet process mixtures. *Journal of Machine Learning Research*, 10:1829–1850, 2009.

P. K. Shivaswamy, C. Bhattacharyya, and A. J. Smola. Second order cone programming approaches for handling missing and uncertain data. *Journal of Machine Learning Research*, 7:1283–1314, 2006.

A. Smola, S. Vishwanathan, and T. Hofmann. Kernel methods for missing variables. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, 2005.

Y. W. Teh, M. J. Beal M. I. Jordan, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581, 2006.

M. E. Tipping. The relevance vector machine. In T. K. Leen S. A. Solla and K. R. Müller, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 12, pages 652–658. MIT Press, 2000.

V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

X. Wang, A. Li, Z. Jiang, and H. Feng. Missing value estimation for DNA microarray gene expression data by support vector regression imputation and orthogonal coding scheme. *BMC Bioinformatics*, 7:32, 2006.

S. R. Waterhouse and A. J. Robinson. Classification using hierarchical mixtures of experts. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing IV*, pages 177–186, 1994.

M. West, P. Müller, and M. D. Escobar. Hierarchical priors and mixture models, with application in regression and density estimation. In P. R. Freeman and A. F. Smith, editors, *Aspects of Uncertainty*, pages 363–386. John Wiley, 1994.

D. Williams and L. Carin. Analytical kernel matrix completion with incomplete multi-view data. In *Proceedings of the International Conference on Machine Learning (ICML) Workshop on Learning with Multiple Views*, pages 80–86, 2005.

D. Williams, X. Liao, Y. Xue, L. Carin, and B. Krishnapuram. On classification with incomplete data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):427–436, 2007.

L. Xu, M. I. Jordan, and G. E. Hinton. An alternative model for mixtures of experts. In *Advances in Neural Information Processing Systems (NIPS) 7*, pages 633–640, 1995.

Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with Dirichlet process priors. *Journal of Machine Learning Research*, 8:35–63, 2007.

K. Yu, A. Schwaighofer, V. Tresp, W.-Y. Ma, and H. Zhang. Collaborative ensemble learning: Combining collaborative and content-based information filtering via hierarchical Bayes. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 616–623, 2003.

J. Zhang, Z. Ghahramani, and Y. Yang. Learning multiple related tasks using latent independent component analysis. In *Advances in Neural Information Processing Systems*, 2006.

Y. Zhang, L. M. Collins, H. Yu, C. Baum, and L. Carin. Sensing of unexploded ordnance with magnetometer and induction data: theory and signal processing. *IEEE Transactions on Geoscience and Remote Sensing*, 41(5):1005–1015, 2003.

X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.