Bayesian Supervised Multilabel Learning with Coupled Embedding and Classification

Mehmet Gönen
Helsinki Institute for Information Technology HIIT
Department of Information and Computer Science
Aalto University School of Science
mehmet.gonen@aalto.fi

Abstract

Coupled training of dimensionality reduction and classification is proposed previously to improve the prediction performance for single-label problems. Following this line of research, in this paper, we introduce a novel *Bayesian supervised multilabel learning* method that combines linear dimensionality reduction with linear binary classification. We present a deterministic variational approximation approach to learn the proposed probabilistic model for multilabel classification. We perform experiments on four benchmark multilabel learning data sets by comparing our method with four baseline linear dimensionality reduction algorithms. Experiments show that the proposed approach achieves good performance values in terms of hamming loss, macro F_1, and micro F_1 on held-out test data. The low-dimensional embeddings obtained by our method are also very useful for exploratory data analysis.

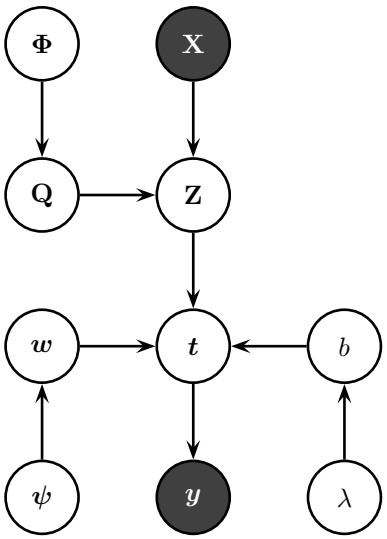
1 Introduction

Multilabel learning considers classification problems where each data point is associated with a set of labels simultaneously instead of just a single label [25]. This setup can be handled by training distinct classifiers for each label separately (i.e., assuming no correlation between the labels). However, exploiting the correlation information between the labels may improve the overall prediction performance. There are two common approaches for exploiting this information: (a) joint learning of the model parameters of distinct classifiers trained for each label [3, 9, 18, 23, 30, 31, 32] and (b) learning a shared subspace and doing classification in this subspace [10, 11, 15, 19, 20, 26, 28, 33]. In this paper, we are focusing on the second approach.

Dimensionality reduction algorithms try to achieve two main goals: (a) removing the inherent noise to improve the prediction performance and (b) obtaining low-dimensional visualizations for exploratory data analysis. *Principal component analysis* (PCA) [16] and *linear discriminant analysis* (LDA) [5] are two well-known algorithms for supervised and unsupervised dimensionality reduction, respectively.

We can use any unsupervised dimensionality reduction algorithm for multilabel learning. However, the key idea in multilabel learning is to use the correlation information between the labels and we only consider supervised dimensionality reduction algorithms. As an early attempt, [28] proposes a supervised *latent semantic indexing* variant that makes use of multiple labels. [15] and [26] modify LDA algorithm for multilabel learning. [20] proposes a probabilistic *canonical correlation analysis* method that can also be applied in semi-supervised settings. [10] and [33] formulate multilabel dimensionality reduction as an eigenvalue problem that uses input features and class labels together.

For supervised learning problems, dimensionality reduction and prediction steps are generally performed separately with two different target functions, leading to low prediction performance. Hence, coupled training of these two steps may improve the overall system performance. Coupled training of the projection matrix and the classifier is studied in the framework of support vector machines by introducing the projection matrix into the optimization problem solved [4, 17]. There are also metric learning methods that are trying to transfer the neighborhood in the input space to the projected subspace in nearest neighbor settings [7,8,27]. [22] uses mixture models for each class to obtain better projections, whereas [13] uses them on both input and output data. The resulting projections found by these approaches are not linear and they can be regarded as manifold learning methods. [29] proposes a supervised probabilistic PCA and an efficient solution method, but the algorithm is developed only for real outputs. [21] formulates a supervised dimensionality reduction algorithm coupled with generalized linear models for binary classification and regression, and maximize a



$$
\begin{aligned}
\phi_s^f &\sim \mathcal{G}(\phi_s^f; \alpha_\phi, \beta_\phi) && \forall (f,s)\\
q_s^f|\phi_s^f &\sim \mathcal{N}(q_s^f; 0, (\phi_s^f)^{-1}) && \forall (f,s)\\
z_i^s|\boldsymbol{q}_s, \boldsymbol{x}_i &\sim \mathcal{N}(z_i^s; \boldsymbol{q}_s^\top \boldsymbol{x}_i, 1) && \forall (s,i)\\
\lambda &\sim \mathcal{G}(\lambda; \alpha_\lambda, \beta_\lambda)\\
b|\lambda &\sim \mathcal{N}(b; 0, \lambda^{-1})\\
\psi_s &\sim \mathcal{G}(\psi_s; \alpha_\psi, \beta_\psi) && \forall s\\
w_s|\psi_s &\sim \mathcal{N}(w_s; 0, \psi_s^{-1}) && \forall s\\
t_i|b, \boldsymbol{w}, \boldsymbol{z}_i &\sim \mathcal{N}(t_i; \boldsymbol{w}^\top \boldsymbol{z}_i + b, 1) && \forall i\\
y_i|t_i &\sim \delta(t_i y_i > 0) && \forall i
\end{aligned}
$$

Figure 1: Bayesian supervised learning with coupled embedding and classification.

target function composed of input and output likelihood terms using an iterative algorithm.

In this paper, we propose a novel *Bayesian supervised multilabel learning* (BSML) method where the linear projection matrix and the binary classification parameters for multilabel learning are learned together to maximize the prediction performance in the projected subspace. We make the following contributions: In Section 2, we give the graphical model of our approach for single-label binary classification and introduce a deterministic variational approximation. Section 3 extends our formulation for multilabel learning and explain the modified variational approximation. We test our algorithms on four different benchmark multilabel data sets in Section 4.

2 Bayesian Supervised Learning with Coupled Embedding and Classification

In order to find a better subspace, we propose to couple dimensionality reduction and binary classification in a joint probabilistic model. The main idea is to map the training instances to a subspace and to perform the classification using the probit model in this projected subspace. Performing dimensionality reduction and classification successively (with two different objective functions) may not result in a predictive subspace and may have low generalization performance. We should consider the predictive performance of the target subspace while learning the projection matrix. Figure 1 illustrates the proposed probabilistic model for binary classification with a graphical model and its distributional assumptions.

The notation we use throughout the manuscript is as follows: The N is the number of training instances. The D shows the dimensionality of the input space and the R gives the dimensionality of the projected subspace. The $D \times N$ data matrix is denoted by $\mathbf{X}$, where the $D \times 1$-dimensional columns of $\mathbf{X}$ by $\boldsymbol{x}_i$. The $D \times R$ matrix of projection variables q_s^f is denoted by $\mathbf{Q}$, where the $D \times 1$-dimensional columns of $\mathbf{Q}$ by $\boldsymbol{q}_s$. The $D \times R$ matrix of priors ϕ_s^f is denoted by $\boldsymbol{\Phi}$, where the $D \times 1$-dimensional columns of $\boldsymbol{\Phi}$ by $\boldsymbol{\phi}_s$. The $R \times N$ matrix of projected variables z_i^s is represented as $\mathbf{Z}$, where the $R \times 1$-dimensional columns of $\mathbf{Z}$ as $\boldsymbol{z}_i$ and the corresponding $N \times 1$-dimensional rows as $\boldsymbol{z}^s$. The $R \times 1$ vector of weight parameters w_s is denoted by $\boldsymbol{w}$. The $R \times 1$ vector of priors ψ_s is denoted by $\boldsymbol{\psi}$. The bias parameter is denoted by b and its prior prior is denoted by λ. The $N \times 1$ vector of auxiliary variables t_i is represented as $\boldsymbol{t}$. The $N \times 1$ vector of associated target values is represented as $\boldsymbol{y}$, where each element $y_i \in \{-1, +1\}$. As short-hand notations, all priors in the model are denoted by $\boldsymbol{\Xi} = \{\lambda, \boldsymbol{\Phi}, \boldsymbol{\psi}\}$, where the remaining variables by $\boldsymbol{\Theta} = \{b, \mathbf{Q}, \boldsymbol{t}, \boldsymbol{w}, \mathbf{Z}\}$ and the hyper-parameters by $\boldsymbol{\omega} = \{\alpha_\lambda, \beta_\lambda, \alpha_\phi, \beta_\phi, \alpha_\psi, \beta_\psi\}$. Dependence on $\boldsymbol{\omega}$ is omitted for clarity throughout the manuscript. $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the normal distribution with the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$. $\mathcal{G}(\cdot; \alpha, \beta)$ denotes the gamma distribution with the shape parameter α and the scale parameter β. $\delta(\cdot)$ denotes the Kronecker delta function that returns 1 if its argument is true and 0 otherwise.

The auxiliary variables between the class labels and the projected instances are introduced to make

the inference procedures efficient [1]. Exact inference for our probabilistic model is intractable and using a Gibbs sampling approach is computationally expensive [6]. We instead formulate a deterministic variational approximation procedure for inference.

The variational methods use a lower bound on the marginal likelihood using an ensemble of factored posteriors to find the joint parameter distribution [2]. Assuming independence between the approximate posteriors in the factorable ensemble can be justified because there is not a strong coupling between our model parameters. We can write the factorable ensemble approximation of the required posterior as

$$p(\boldsymbol{\Theta}, \boldsymbol{\Xi}|\mathbf{X}, \boldsymbol{y}) \approx q(\boldsymbol{\Theta}, \boldsymbol{\Xi}) = q(\boldsymbol{\Phi})q(\mathbf{Q})q(\mathbf{Z}) \\ q(\lambda)q(\boldsymbol{\psi})q(b, \boldsymbol{w})q(\boldsymbol{t})$$

and define each factor in the ensemble just like its full conditional distribution:

$$\begin{aligned} q(\boldsymbol{\Phi}) &= \prod_{f=1}^{D}\prod_{s=1}^{R} \mathcal{G}(\phi_s^f; \alpha(\phi_s^f), \beta(\phi_s^f)) \\ q(\mathbf{Q}) &= \prod_{s=1}^{R} \mathcal{N}(\boldsymbol{q}_s; \mu(\boldsymbol{q}_s), \Sigma(\boldsymbol{q}_s)) \\ q(\mathbf{Z}) &= \prod_{i=1}^{N} \mathcal{N}(\boldsymbol{z}_i; \mu(\boldsymbol{z}_i), \Sigma(\boldsymbol{z}_i)) \\ q(\lambda) &= \mathcal{G}(\lambda; \alpha(\lambda), \beta(\lambda)) \\ q(\boldsymbol{\psi}) &= \prod_{s=1}^{R} \mathcal{G}(\psi_s; \alpha(\psi_s), \beta(\psi_s)) \\ q(b, \boldsymbol{w}) &= \mathcal{N}\left(\begin{bmatrix} b \\ \boldsymbol{w} \end{bmatrix}; \mu(b, \boldsymbol{w}), \Sigma(b, \boldsymbol{w})\right) \\ q(\boldsymbol{t}) &= \prod_{i=1}^{N} \mathcal{TN}(t_i; \mu(t_i), \Sigma(t_i), \rho(t_i)) \end{aligned}$$

where $\alpha(\cdot)$, $\beta(\cdot)$, $\mu(\cdot)$, and $\Sigma(\cdot)$ denote the shape parameter, the scale parameter, the mean vector, and the covariance matrix for their arguments, respectively. $\mathcal{TN}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \rho(\cdot))$ denotes the truncated normal distribution with the mean vector $\boldsymbol{\mu}$, the covariance matrix $\boldsymbol{\Sigma}$, and the truncation rule $\rho(\cdot)$ such that $\mathcal{TN}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \rho(\cdot)) \propto \mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ if $\rho(\cdot)$ is true and $\mathcal{TN}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \rho(\cdot)) = 0$ otherwise.

We can bound the marginal likelihood using Jensen's inequality:

$$\log p(\boldsymbol{y}|\mathbf{X}) \geq \\ \mathrm{E}_{q(\boldsymbol{\Theta},\boldsymbol{\Xi})}[\log p(\boldsymbol{y}, \boldsymbol{\Theta}, \boldsymbol{\Xi}|\mathbf{X})] - \mathrm{E}_{q(\boldsymbol{\Theta},\boldsymbol{\Xi})}[\log q(\boldsymbol{\Theta}, \boldsymbol{\Xi})] \tag{2.1}$$

and optimize this bound by optimizing with respect to each factor separately until convergence. The approximate posterior distribution of a specific factor $\boldsymbol{\tau}$ can be found as

$$q(\boldsymbol{\tau}) \propto \exp\big(\mathrm{E}_{q(\{\boldsymbol{\Theta},\boldsymbol{\Xi}\}\setminus\boldsymbol{\tau})}[\log p(\boldsymbol{y}, \boldsymbol{\Theta}, \boldsymbol{\Xi}|\mathbf{X})]\big).$$

For our model, thanks to the conjugacy, the resulting approximate posterior distribution of each factor follows the same distribution as the corresponding factor.

2.1 Inference Details The approximate posterior distribution of the priors of the precisions for the projection matrix can be found as a product of gamma distributions:

$$q(\boldsymbol{\Phi}) = \prod_{f=1}^{D}\prod_{s=1}^{R} \mathcal{G}\left(\phi_s^f; \alpha_\phi + \frac{1}{2}, \Big(\frac{1}{\beta_\phi} + \frac{\widetilde{(q_s^f)^2}}{2}\Big)^{-1}\right)$$

where the tilde notation denotes the posterior expectations as usual, i.e., $\widetilde{f(\boldsymbol{\tau})} = \mathrm{E}_{q(\boldsymbol{\tau})}[f(\boldsymbol{\tau})]$. The approximate posterior distribution of the projection matrix is a product of multivariate normal distributions:

$$q(\mathbf{Q}) = \prod_{s=1}^{R} \mathcal{N}(\boldsymbol{q}_s; \Sigma(\boldsymbol{q}_s)\mathbf{X}\widetilde{\boldsymbol{z}^s}, (\mathrm{diag}(\widetilde{\boldsymbol{\phi_s}}) + \mathbf{X}\mathbf{X}^\top)^{-1}).$$

The approximate posterior distribution of the projected instances can also be formulated as a product of multivariate normal distributions:

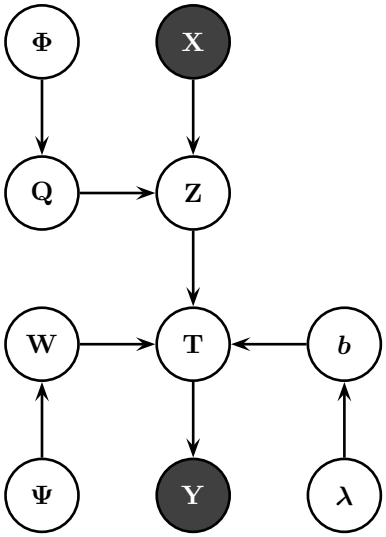
$$q(\mathbf{Z}) = \prod_{i=1}^{N} \mathcal{N}(\boldsymbol{z}_i; \Sigma(\boldsymbol{z}_i)(\widetilde{\mathbf{Q}^\top}\boldsymbol{x}_i + \widetilde{\boldsymbol{w}}\widetilde{t_i} - \widetilde{\boldsymbol{w}}b), \\ (\mathbf{I} + \widetilde{\boldsymbol{w}\boldsymbol{w}^\top})^{-1}).$$

The approximate posterior distributions of the priors on the bias and the weight vector can be found in terms of gamma distributions:

$$\begin{aligned} q(\lambda) &= \mathcal{G}\left(\lambda; \alpha_\lambda + \frac{1}{2}, \Big(\frac{1}{\beta_\lambda} + \frac{\widetilde{b^2}}{2}\Big)^{-1}\right) \\ q(\boldsymbol{\psi}) &= \prod_{s=1}^{R} \mathcal{G}\left(\psi_s; \alpha_\psi + \frac{1}{2}, \Big(\frac{1}{\beta_\psi} + \frac{\widetilde{w_s^2}}{2}\Big)^{-1}\right). \end{aligned}$$

The approximate posterior distribution of the classification parameters is a product of multivariate normal distributions:

$$q(b, \boldsymbol{w}) = \mathcal{N}\left(\begin{bmatrix} b \\ \boldsymbol{w} \end{bmatrix}; \Sigma(b, \boldsymbol{w}) \begin{bmatrix} \mathbf{1}^\top\widetilde{\boldsymbol{t}} \\ \widetilde{\mathbf{Z}}\widetilde{\boldsymbol{t}} \end{bmatrix}, \\ \begin{bmatrix} \widetilde{\lambda} + N & \mathbf{1}^\top\widetilde{\mathbf{Z}^\top} \\ \widetilde{\mathbf{Z}}\mathbf{1} & \mathrm{diag}(\widetilde{\boldsymbol{\psi}}) + \widetilde{\mathbf{Z}\mathbf{Z}^\top} \end{bmatrix}^{-1}\right).$$



$$
\begin{aligned}
\phi_s^f &\sim \mathcal{G}(\phi_s^f; \alpha_\phi, \beta_\phi) && \forall (f,s)\\
q_s^f|\phi_s^f &\sim \mathcal{N}(q_s^f; 0, (\phi_s^f)^{-1}) && \forall (f,s)\\
z_i^s|\boldsymbol{q}_s, \boldsymbol{x}_i &\sim \mathcal{N}(z_i^s; \boldsymbol{q}_s^\top \boldsymbol{x}_i, 1) && \forall (s,i)\\
\lambda_o &\sim \mathcal{G}(\lambda_o; \alpha_\lambda, \beta_\lambda) && \forall o\\
b_o|\lambda_o &\sim \mathcal{N}(b_o; 0, \lambda_o^{-1}) && \forall o\\
\psi_o^s &\sim \mathcal{G}(\psi_o^s; \alpha_\psi, \beta_\psi) && \forall (s,o)\\
w_o^s|\psi_o^s &\sim \mathcal{N}(w_o^s; 0, (\psi_o^s)^{-1}) && \forall (s,o)\\
t_i^o|b_o, \boldsymbol{w}_o, \boldsymbol{z}_i &\sim \mathcal{N}(t_i^o; \boldsymbol{w}_o^\top \boldsymbol{z}_i + b_o, 1) && \forall (o,i)\\
y_i^o|t_i^o &\sim \delta(t_i^o y_i^o > 0) && \forall (o,i)
\end{aligned}
$$

Figure 2: Bayesian supervised multilabel learning with coupled embedding and classification.

The approximate posterior distribution of the auxiliary variables is a product of truncated normal distributions:

$$q(\boldsymbol{t}) = \prod_{i=1}^{N} \mathcal{TN}(t_i; \widetilde{\boldsymbol{w}^\top} \widetilde{\boldsymbol{z}}_i + \widetilde{b}, 1, t_i y_i > 0)$$

where we need to find the posterior expectations in order to update the approximate posterior distributions of the projected instances and the classification parameters. Fortunately, the truncated normal distribution has a closed-form formula for its expectation.

2.2 Convergence The inference mechanism sequentially updates the approximate posterior distributions of the model parameters and the latent variables until convergence, which can be checked by calculating the lower bound in (2.1). The first term of the lower bound corresponds to the sum of exponential form expectations of the distributions in the joint likelihood. The second term is the sum of negative entropies of the approximate posteriors in the ensemble. The only nonstandard distribution in the second term is the truncated normal distributions of the auxiliary variables; nevertheless, the truncated normal distribution has a closed-form formula also for its entropy.

2.3 Prediction In the prediction step, we can replace $p(\mathbf{Q}|\mathbf{X}, \boldsymbol{y})$ with its approximate posterior distribution $q(\mathbf{Q})$ and obtain the predictive distribution of the projected instance $\boldsymbol{z}_\star$ for a new data point $\boldsymbol{x}_\star$ as

$$p(\boldsymbol{z}_\star|\boldsymbol{x}_\star, \mathbf{X}, \boldsymbol{y}) = \prod_{s=1}^{R} \mathcal{N}(z_\star^s; \mu(\boldsymbol{q}_s)^\top \boldsymbol{x}_\star, 1 + \boldsymbol{x}_\star^\top \Sigma(\boldsymbol{q}_s) \boldsymbol{x}_\star).$$

The predictive distribution of the auxiliary variable $t_\star$ can also be found by replacing $p(b, \boldsymbol{w}|\mathbf{X}, \boldsymbol{y})$ with its approximate posterior distribution $q(b, \boldsymbol{w})$:

$$p(t_\star|\mathbf{X}, \boldsymbol{y}, \boldsymbol{z}_\star) = \mathcal{N}\left(t_\star; \mu(b, \boldsymbol{w})^\top \begin{bmatrix} 1 \\ \boldsymbol{z}_\star \end{bmatrix}, 1 + \begin{bmatrix} 1 & \boldsymbol{z}_\star \end{bmatrix} \Sigma(b, \boldsymbol{w}) \begin{bmatrix} 1 \\ \boldsymbol{z}_\star \end{bmatrix}\right)$$

and the predictive distribution of the class label $y_\star$ can be formulated using the auxiliary variable distribution:

$$p(y_\star = +1|\boldsymbol{x}_\star, \mathbf{X}, \boldsymbol{y}) = \Phi\left(\frac{\mu(t_\star)}{\Sigma(t_\star)}\right)$$

where $\Phi(\cdot)$ is the standardized normal cumulative distribution function.

3 Bayesian Supervised Multilabel Learning with Coupled Embedding and Classification

We use the probabilistic model in the previous section as our base model and extend this model for multilabel learning. The training instances are again mapped to a subspace before classification. In order to benefit from the correlation between the class labels, we assume a common subspace and perform classification for all labels in that subspace using different classifiers for each label separately. The predictive quality of the subspace now depends on the prediction performances for multiple labels instead of a single one. Figure 2 illustrates the modified probabilistic model for multilabel binary classification with a graphical model and its distributional assumptions.

There are slight modifications in the notation we described previously: The L is the number of labels associated with each data instance. The $R \times L$ matrix of weight parameters w_o^s is denoted by $\mathbf{W}$, where the $R \times 1$-dimensional columns of $\mathbf{W}$ by $\boldsymbol{w}_o$. The $R \times L$ matrix of priors ψ_o^s is denoted by $\boldsymbol{\Psi}$, where the $R \times 1$-dimensional columns of $\boldsymbol{\Psi}$ by $\boldsymbol{\psi}_o$. The $L \times 1$ vector of bias parameters b_o is denoted by $\boldsymbol{b}$. The $L \times 1$ vector of priors λ_o is denoted by $\boldsymbol{\lambda}$. The $L \times N$ matrix of auxiliary variables t_i^o is represented as $\mathbf{T}$, where the $N \times 1$-dimensional rows of $\mathbf{T}$ as $\boldsymbol{t}^o$. The $L \times N$ matrix of associated target values is represented as $\mathbf{Y}$, where each element $y_i^o \in \{-1, +1\}$. All priors in the model are denoted by $\boldsymbol{\Xi} = \{\boldsymbol{\lambda}, \boldsymbol{\Phi}, \boldsymbol{\Psi}\}$, where the remaining variables by $\boldsymbol{\Theta} = \{\boldsymbol{b}, \mathbf{Q}, \mathbf{T}, \mathbf{W}, \mathbf{Z}\}$.

There is not a strong coupling between our model parameters as before and we can write the factorable ensemble approximation of the required posterior as

$$p(\boldsymbol{\Theta}, \boldsymbol{\Xi}|\mathbf{X}, \mathbf{Y}) \approx q(\boldsymbol{\Theta}, \boldsymbol{\Xi}) = q(\boldsymbol{\Phi})q(\mathbf{Q})q(\mathbf{Z}) \\ q(\boldsymbol{\lambda})q(\boldsymbol{\Psi})q(\boldsymbol{b}, \mathbf{W})q(\mathbf{T})$$

where only the last four terms are modified to handle multilabel learning:

$$q(\boldsymbol{\lambda}) = \prod_{o=1}^{L} \mathcal{G}(\lambda_o; \alpha(\lambda_o), \beta(\lambda_o))$$

$$q(\boldsymbol{\Psi}) = \prod_{s=1}^{R} \prod_{o=1}^{L} \mathcal{G}(\psi_o^s; \alpha(\psi_o^s), \beta(\psi_o^s))$$

$$q(\boldsymbol{b}, \mathbf{W}) = \prod_{o=1}^{L} \mathcal{N}\left(\begin{bmatrix} b_o \\ \boldsymbol{w}_o \end{bmatrix}; \mu(b_o, \boldsymbol{w}_o), \Sigma(b_o, \boldsymbol{w}_o) \right)$$

$$q(\mathbf{T}) = \prod_{o=1}^{L} \prod_{i=1}^{N} \mathcal{TN}(t_i^o; \mu(t_i^o), \Sigma(t_i^o), \rho(t_i^o)).$$

We can again bound the marginal likelihood using Jensen's inequality:

$$\log p(\mathbf{Y}|\mathbf{X}) \geq \\ \mathrm{E}_{q(\boldsymbol{\Theta},\boldsymbol{\Xi})}[\log p(\mathbf{Y}, \boldsymbol{\Theta}, \boldsymbol{\Xi}|\mathbf{X})] - \mathrm{E}_{q(\boldsymbol{\Theta},\boldsymbol{\Xi})}[\log q(\boldsymbol{\Theta}, \boldsymbol{\Xi})] \tag{3.2}$$

and optimize this bound by optimizing with respect to each factor separately until convergence. The approximate posterior distribution of a specific factor $\boldsymbol{\tau}$ can also be found as

$$q(\boldsymbol{\tau}) \propto \exp\big(\mathrm{E}_{q(\{\boldsymbol{\Theta},\boldsymbol{\Xi}\}\setminus\boldsymbol{\tau})}[\log p(\mathbf{Y}, \boldsymbol{\Theta}, \boldsymbol{\Xi}|\mathbf{X})]\big).$$

3.1 Inference Details The approximate posterior distribution of the projected instances can be formulated as a product of multivariate normal distributions:

$$q(\mathbf{Z}) = \prod_{i=1}^{N} \mathcal{N}\Bigg(\boldsymbol{z}_i; \Sigma(\boldsymbol{z}_i)\Big(\widetilde{\mathbf{Q}^\top} \boldsymbol{x}_i + \sum_{o=1}^{L} \big(\widetilde{\boldsymbol{w}_o} \widetilde{t_i^o} - \widetilde{\boldsymbol{w}_o b_o} \big) \Big), \\ \Big(\mathbf{I} + \sum_{o=1}^{L} \widetilde{\boldsymbol{w}_o \boldsymbol{w}_o^\top} \Big)^{-1} \Bigg)$$

where the classification parameters and the auxiliary variables defined for each label are used together.

The approximate posterior distributions of the priors on the biases and the weight vectors can be found as products of gamma distributions:

$$q(\boldsymbol{\lambda}) = \prod_{o=1}^{L} \mathcal{G}\Big(\lambda_o; \alpha_\lambda + \frac{1}{2}, \big(\frac{1}{\beta_\lambda} + \frac{\widetilde{b_o^2}}{2} \big)^{-1} \Big)$$

$$q(\boldsymbol{\Psi}) = \prod_{s=1}^{R} \prod_{o=1}^{L} \mathcal{G}\Big(\psi_o^s; \alpha_\psi + \frac{1}{2}, \big(\frac{1}{\beta_\psi} + \frac{\widetilde{(w_o^s)^2}}{2} \big)^{-1} \Big).$$

The approximate posterior distribution of the classification parameters is a product of multivariate normal distributions:

$$q(\boldsymbol{b}, \mathbf{W}) = \prod_{o=1}^{L} \mathcal{N}\Bigg(\begin{bmatrix} b_o \\ \boldsymbol{w}_o \end{bmatrix}; \Sigma(b_o, \boldsymbol{w}_o) \begin{bmatrix} \mathbf{1}^\top \widetilde{\boldsymbol{t}^o} \\ \widetilde{\mathbf{Z}\boldsymbol{t}^o} \end{bmatrix}, \\ \begin{bmatrix} \widetilde{\lambda_o} + N & \mathbf{1}^\top \widetilde{\mathbf{Z}^\top} \\ \widetilde{\mathbf{Z}\mathbf{1}} & \mathrm{diag}(\widetilde{\boldsymbol{\psi}_o}) + \widetilde{\mathbf{Z}\mathbf{Z}^\top} \end{bmatrix}^{-1} \Bigg).$$

The approximate posterior distribution of the auxiliary variables is a product of truncated normal distributions:

$$q(\mathbf{T}) = \prod_{o=1}^{L} \prod_{i=1}^{N} \mathcal{TN}(t_i^o; \widetilde{\boldsymbol{w}_o^\top} \widetilde{\boldsymbol{z}_i} + \widetilde{b_o}, 1, t_i^o y_i^o > 0).$$

3.2 Convergence The inference mechanism is very similar to the base model of the previous section and the lower bound can also be calculated similarly using (3.2). Exact form of the variational lower bound can be found in Appendix A.

3.3 Prediction The predictive distribution of the auxiliary variable $t_\star^o$ can be formulated as

$$p(t_\star^o|\mathbf{X}, \mathbf{Y}, \boldsymbol{z}_\star) = \\ \mathcal{N}\Bigg(t_\star^o; \mu(b_o, \boldsymbol{w}_o)^\top \begin{bmatrix} 1 \\ \boldsymbol{z}_\star \end{bmatrix}, 1 + \begin{bmatrix} 1 & \boldsymbol{z}_\star \end{bmatrix} \Sigma(b_o, \boldsymbol{w}_o) \begin{bmatrix} 1 \\ \boldsymbol{z}_\star \end{bmatrix} \Bigg)$$

and the predictive distribution of the class label $y_\star^o$ can be found as

$$p(y_\star^o = +1|\boldsymbol{x}_\star, \mathbf{X}, \mathbf{Y}) = \Phi\Big(\frac{\mu(t_\star^o)}{\Sigma(t_\star^o)} \Big).$$

3.4 Computational Complexity Updating the projection matrix $\mathbf{Q}$ is the most time-consuming step, which requires inverting $D \times D$ matrices for the covariance calculations and dominates the overall running time. When D is very large, the dimensionality of the input space should be reduced using an unsupervised dimensionality reduction method (e.g., PCA) before running the algorithm.

3.5 Inherent Regularization The multiplication of the projection matrix $\mathbf{Q}$ and the supervised learning parameters $\mathbf{W}$ can be interpreted as the model parameters of linear classifiers for the original representation. However, if $L > R$, the parameter matrix $\mathbf{QW}$ is guaranteed to be low-rank due to this decomposition leading to a more regularized solution. For multivariate regression estimation, our model can be interpreted as a full Bayesian treatment of reduced-rank regression [24].

3.6 Effect of Priors The precision priors for the projection matrix can be modified to decide which features should be used or to determine the dimensionality automatically when generating the projected instances. Using row-wise sparse priors instead of entry-wise priors on the projection matrix leads to feature selection, whereas using column-wise sparse priors on the projection matrix enables us to determine the dimensionality of the projected subspace.

4 Experiments

We test our new algorithm BSML on four different data sets by comparing it with four (one unsupervised and three supervised) baseline dimensionality reduction algorithms, namely, PCA [16], *multilabel dimensionality reduction via dependency maximization* (MDDM) [33], *multilabel least squares* (MLLS) [10], and *multilabel linear discriminant analysis* (MLDA) [26]. BSML combines dimensionality reduction and binary classification for multilabel learning in a joint framework. In order to have comparable algorithms, we perform binary classification using *probit model* (PROBIT) on each label separately, after reducing dimensionality using baseline algorithms. The suffix +PROBIT corresponds to learning a binary classifier for each label in the projected subspace using PROBIT. We also report the classification results obtained by training a PROBIT on each label separately without dimensionality reduction to see the baseline performance.

We implement variational approximation methods for both PROBIT and BSML in Matlab, where we take 500 iterations. These implementations are publicly available at `http://users.ics.tkk.fi/gonen/bsml/`. The default hyper-parameter values for PROBIT and BSML are selected as $(\alpha_\lambda, \beta_\lambda, \alpha_\psi, \beta_\psi) = (1, 1, 1, 1)$ and $(\alpha_\lambda, \beta_\lambda, \alpha_\phi, \beta_\phi, \alpha_\psi, \beta_\psi) = (1, 1, 1, 1, 1, 1)$, respectively. We implement our own versions for PCA, MDDM, MLLS, and MLDA. We use the provided default parameter values for MDDM, MLLS, and MLDA.

We use four widely used benchmark data sets, namely, `Emotions`, `Medical`, `Scene`, and `Yeast`, from different domains to compare our algorithm with the baseline algorithms using provided train/test splits. These data sets are publicly available at `http://mulan.sourceforge.net/datasets.html` and their characteristics are summarized in Table 1.

Table 1: Summary of data set characteristics.

Data Set	Domain	$N_{\texttt{train}}$	$N_{\texttt{test}}$	D	L
`Emotions`	music	391	202	72	6
`Medical`	text	333	645	1449	45
`Scene`	image	1211	1196	294	6
`Yeast`	biology	1500	917	103	14

Three popular performance measures for multilabel learning, namely, *hamming loss*, *macro* F_1, and *micro* F_1 are used to compare the algorithms. Hamming loss is the average classification error over the labels. The smaller the value of hamming loss, the better the performance. Macro F_1 is the average of F_1 scores over the labels. The larger the value of macro F_1, the better the performance. Micro F_1 calculates the F_1 score over the labels as a whole. The larger the value of micro F_1, the better the performance.

Figure 3 gives the classification results on `Emotions` data set. We perform experiments with $R = 1, 2, \ldots, 6$ for all of the methods except MLDA and with $R = 1, 2, \ldots, 5$ for MLDA. Note that the dimensionality of the projected subspace can be at most D for PCA, $L-1$ for MLDA, and L for MDDM and MLLS. There is not such a restriction for BSML. We see that BSML clearly outperforms all of the baseline algorithms for all of the dimensions tried in terms of hamming loss and the performance difference is around two per cent when $R = 2$. BSML results with all of the dimensions tried are better than using the original feature representation without any dimensionality reduction (i.e., PROBIT). However, BSML and MLDA+PROBIT seem comparable in terms of macro F_1 and micro F_1. Both algorithms achieve similar macro F_1 and micro F_1 values as PROBIT using two or more dimensions.

Figure 4 shows the classification results on `Medical` data set. We perform experiments with $R = 1, 2, \ldots, 10$ for all of the methods. We see that BSML clearly out-

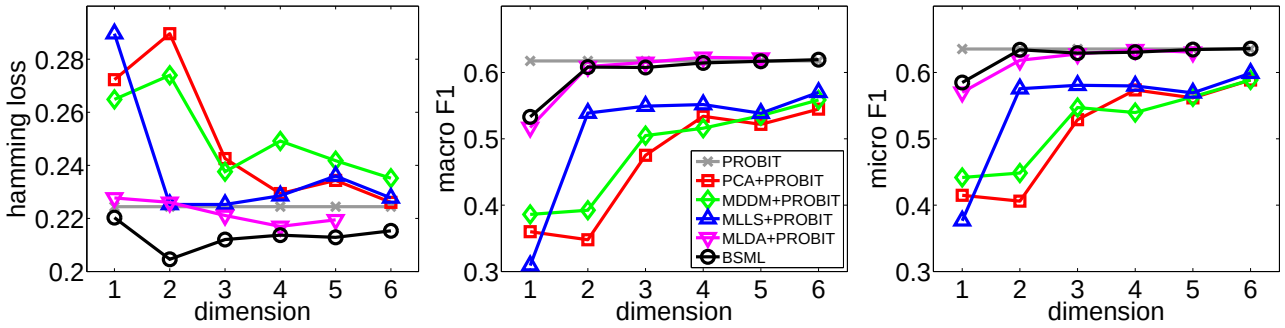


Figure 3: Comparison of algorithms on `Emotions` data set.

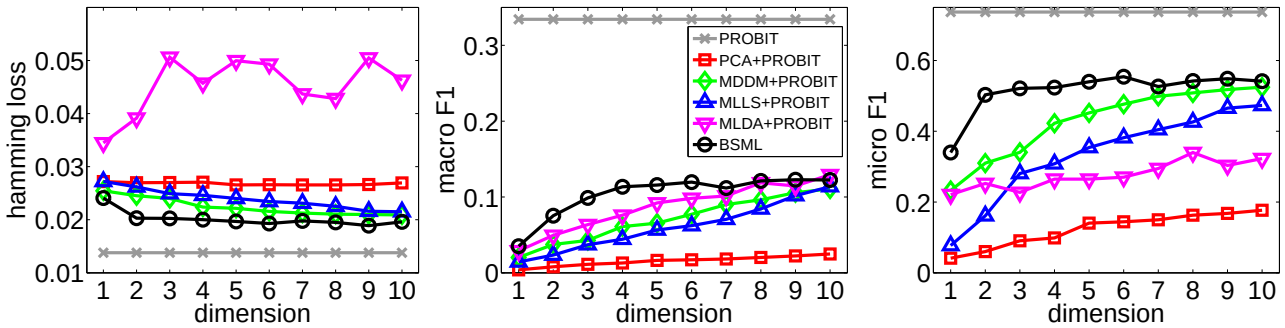


Figure 4: Comparison of algorithms on `Medical` data set.

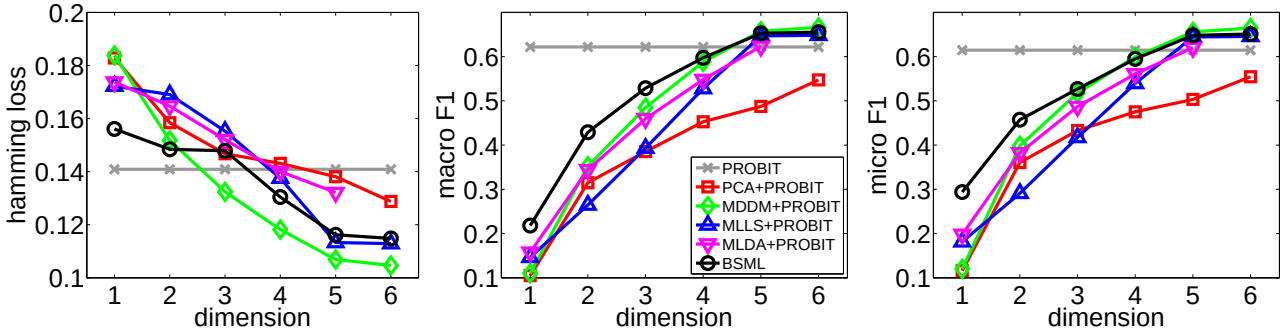


Figure 5: Comparison of algorithms on `Scene` data set.

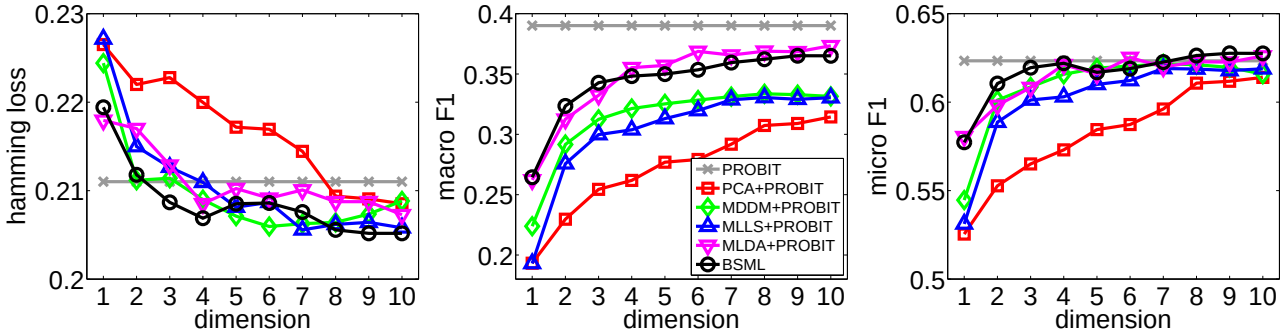


Figure 6: Comparison of algorithms on `Yeast` data set.

performs all of the dimensionality reduction algorithms in terms of three performance measures. However, the performance values are not as good as PROBIT due to the high dimensionality (i.e., $D = 1449$) and the large number class labels (i.e., $L = 45$). The performance difference between PROBIT and BSML in terms of hamming loss is less than one per cent with only two dimensions. The differences between BSML and the other dimensionality reduction algorithms are more significant when the projected subspace is very low-dimensional.

The classification results on `Scene` data set are given in Figure 5. We perform experiments with $R = 1, 2, \ldots, 6$ for all of the methods except MLDA and with $R = 1, 2, \ldots, 5$ for MLDA. In terms of hamming loss, BSML is better than other dimensionality reduction algorithms with $R = 1$ and 2. However, MDDM+PROBIT achieves lower hamming loss values after two dimensions. All of the dimensionality reduction algorithms get lower hamming loss values than PROBIT after four dimensions. In terms of macro F_1 and micro F_1, BSML is the best algorithm among dimensionality reduction methods up to four dimensions. After four dimensions, MDDM+PROBIT, BSML, MLLS+PROBIT, and MLDA+PROBIT are better than PROBIT in terms of both macro F_1 and micro F_1.

The classification results on `Yeast` data set are shown in Figure 6. We perform experiments with $R = 1, 2, \ldots, 10$ for all of the methods. MDDM+PROBIT is the best algorithm in terms of hamming loss for $R = 2$, whereas BSML is the best one for $R = 3$ and 4. After four dimensions, there is no clear outperforming algorithm. When the projected subspace is one-, two-, or three-dimensional, BSML is clearly better than all of the dimensionality reduction algorithms in terms of macro F_1 and micro F_1.

We use PROBIT to classify projected instances for comparing BSML with baseline dimensionality reduction algorithms in terms of classification performance. This may add some bias to the comparisons because BSML contains PROBIT in its formulation. We also replicate the experiments using *k-nearest neighbor* as the classification algorithm after dimensionality reduction. The classification performances on the four data sets are very similar to the ones obtained using PROBIT. This shows that the superiority of BSML especially on very low dimensions can not be explained by the use of PROBIT only.

In addition to performing classification, the projected subspace found by BSML can also be used for exploratory data analysis. Figures 7 and 8 show two-dimensional embeddings of training data points and classification boundaries for each label obtained by MLDA and BSML on `Emotions` data set. The class labels of this data set corresponds to different emotions assigned to musical pieces by three experts. We can see that, with two dimensions, BSML achieves to embed data points in a more predictive subspace than MLDA. The correlations between different labels are clearly visible in the embedding obtained, for example, the positive correlation between labels `quiet-still` and `sad-lonely` and the negative correlation between labels `relaxing-calm` and `angry-fearful`.

5 Discussion

We present a Bayesian supervised multilabel learning method that couples linear dimensionality reduction and linear binary classification. We provide detailed derivations for supervised and semi-supervised learning using a deterministic variational approximation approach. Experimental results on four benchmark multilabel learning data sets show that our model obtains better performance values than baseline linear dimensionality reduction algorithms most of the time. The low-dimensional embeddings obtained by our method can also be used for exploratory data analysis.

The proposed model can be extended in different directions: First, we can modify the priors on the projection matrix in order to determine the dimensionality of the projected subspace automatically. Using column-wise priors instead of entry-wise priors allows us to discard unnecessary dimensions (i.e., automatic relevance determination) [14]. Second, we can make use of unlabeled data points in addition to labeled ones (i.e., semi-supervised learning) assuming a low-density region between the classes [12]. Lastly, we can learn a unified subspace for multiple input representations (i.e., multitask learning) by exploiting the correlations between different tasks defined on different input features. This extension also allows us to learn a transfer function between different feature representations (i.e., transfer learning).

References

[1] James H. Albert and Siddhartha Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88:669–679, 1993.

[2] Matthew J. Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, The Gatsby Computational Neuroscience Unit, University College London, 2003.

[3] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37:1757–1771, 2004.

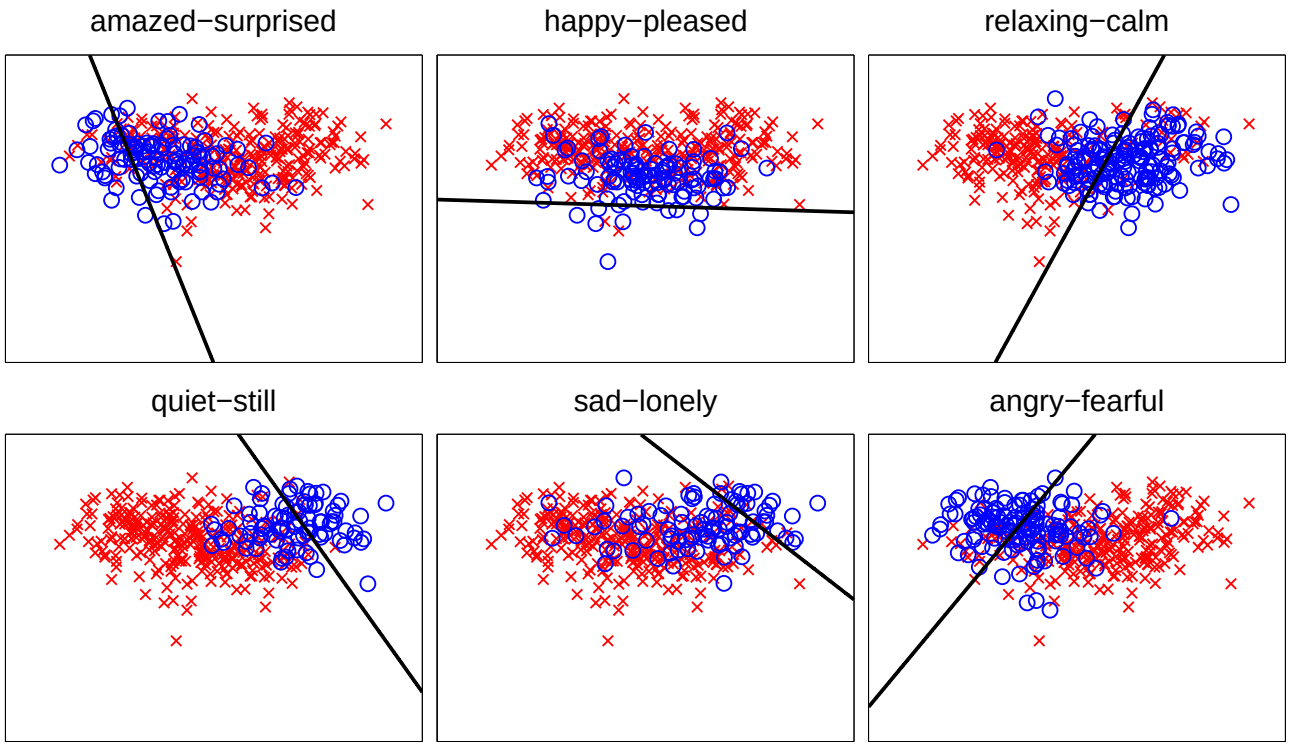


Figure 7: Two-dimensional embedding obtained by MLDA on `Emotions` data set.

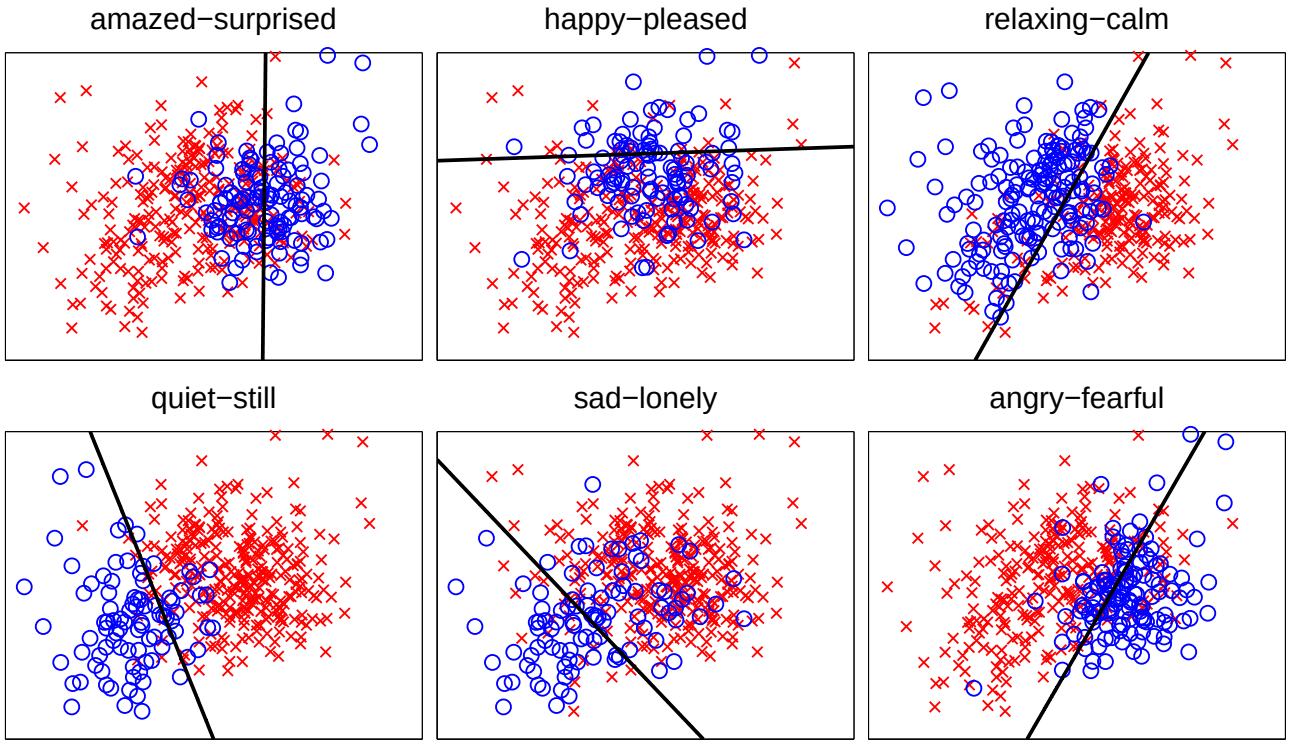


Figure 8: Two-dimensional embedding obtained by BSML on `Emotions` data set.

[4] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46:131–159, 2002.

[5] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7 Part II:179–188, 1936.

[6] A. E. Gelfand and A. F. M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409, 1990.

[7] Amir Globerson and Sam Roweis. Metric learning by collapsing classes. In *Advances in Neural Information Processing Systems 18*, 2006.

[8] Jacob Goldberger, Sam Roweis, Geoff Hinton, and Ruslan Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems 17*, 2005.

[9] Yuhong Guo and Suicheng Gu. Multi-label classification using conditional dependency networks. In *Proceedings of the 22nd International Joint Conference on Artifical Intelligence*, 2011.

[10] Shuiwang Ji, Lei Tang, Shipeng Yu, and Jieping Ye. A shared-subspace learning framework for multi-label classification. *ACM Transactions on Knowledge Discovery from Data*, 4(2):8:1–8:29, 2010.

[11] Shuiwang Ji and Jieping Ye. Linear dimensionality reduction for multi-label classification. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence*, 2009.

[12] Neil D. Lawrence and Michael I. Jordan. Semi-supervised learning via Gaussian processes. In *Advances in Neural Information Processing Systems 17*, 2005.

[13] Kai Mao, Feng Liang, and Sayan Mukherjee. Supervised dimension reduction using Bayesian mixture modeling. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 2010.

[14] Radford M. Neal. *Bayesian Learning for Neural Networks*. Lecture Notes in Statistics 118. Springer, 1996.

[15] Cheong Hee Park and Moonhwi Lee. On applying linear discriminant analysis for multi-labeled problems. *Pattern Recognition Letters*, 29:878–887, 2008.

[16] Karl Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

[17] Francisco Pereira and Geoffrey Gordon. The support vector decomposition machine. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

[18] James Petterson and Tiberio Caetano. Reverse multi-label learning. In *Advances in Neural Information Processing Systems 23*, 2010.

[19] Buyue Qian and Ian Davidson. Semi-supervised dimension reduction for multi-label classification. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 2010.

[20] Piyush Rai and Hal Daumé III. Multi-label prediction via sparse infinite CCA. In *Advances in Neural Information Processing Systems 22*, 2009.

[21] Irina Rish, Genady Grabarnik, Guillermo Cecchi, Francisco Pereira, and Geoffrey J. Gordon. Closed-form supervised dimensionality reduction with generalized linear models. In *Proceedings of the 25th International Conference on Machine Learning*, 2008.

[22] Sajama and Alon Orlitsky. Supervised dimensionality reduction using mixture models. In *Proceedings of the 22nd International Conference on Machine Learning*, 2005.

[23] Liang Sun, Shuiwang Ji, and Jieping Ye. Hypergraph spectral learning for multi-label classification. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008.

[24] Michael K.-S. Tso. Reduced-rank regression and canonical analysis. *Journal of the Royal Statistical Society: Series B (Methodological)*, 43:183–189, 1981.

[25] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In Oded Maimon and Lior Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 667–685. Springer, 2009.

[26] Hua Wang, Chris Ding, and Heng Huang. Multi-label linear discriminant analysis. In *Proceedings of the 11th European Conference on Computer Vision*, 2010.

[27] Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

[28] Kai Yu, Shipeng Yu, and Volker Tresp. Multi-label informed latent semantic indexing. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2005.

[29] Shipeng Yu, Kai Yu, Volker Tresp, Hans-Peter Kriegel, and Mingrui Wu. Supervised probabilistic principal component analysis. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.

[30] Min-Ling Zhang. LIFT: Multi-label learning with label-specific features. In *Proceedings of the 22nd International Joint Conference on Artifical Intelligence*, 2011.

[31] Min-Ling Zhang and Zhi-Hua Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40:2038–2048, 2007.

[32] Wei Zhang, Xiangyang Xue, Jianping Fan, Xiaojing Huang, Bin Wu, and Mingjie Liu. Multi-kernel multi-label learning with max-margin concept network. In *Proceedings of the 22nd International Joint Conference on Artifical Intelligence*, 2011.

[33] Yin Zhang and Zhi-Hua Zhou. Multilabel dimensionality reduction via dependence maximization. *ACM Transactions on Knowledge Discovery from Data*, 4(3):14:1–14:21, 2010.

A Variational Lower Bound for Multilabel Learning

The variational lower bound of our multilabel learning model can be written as

$$\mathcal{L} = \mathrm{E}_{q(\boldsymbol{\Theta},\boldsymbol{\Xi})}[\log p(\mathbf{Y},\boldsymbol{\Theta},\boldsymbol{\Xi}|\mathbf{X})] - \mathrm{E}_{q(\boldsymbol{\Theta},\boldsymbol{\Xi})}[\log q(\boldsymbol{\Theta},\boldsymbol{\Xi})]$$

where the joint likelihood is defined as

$$p(\mathbf{Y},\boldsymbol{\Theta},\boldsymbol{\Xi}|\mathbf{X}) = p(\boldsymbol{\Phi})p(\mathbf{Q}|\boldsymbol{\Phi})p(\mathbf{Z}|\mathbf{Q},\mathbf{X})p(\boldsymbol{\lambda})p(b|\boldsymbol{\lambda}) \\ p(\boldsymbol{\Psi})p(\mathbf{W}|\boldsymbol{\Psi})p(\mathbf{T}|\boldsymbol{b},\mathbf{W},\mathbf{Z})p(\mathbf{Y}|\mathbf{T}).$$

Using these definitions, the variational lower bound becomes

$$\begin{aligned}\mathcal{L} = {} & \mathrm{E}_{q(\boldsymbol{\Phi})}[\log p(\boldsymbol{\Phi})] + \mathrm{E}_{q(\boldsymbol{\Phi})q(\mathbf{Q})}[\log p(\mathbf{Q}|\boldsymbol{\Phi})] \\ & + \mathrm{E}_{q(\mathbf{Q})q(\mathbf{Z})}[\log p(\mathbf{Z}|\mathbf{Q},\mathbf{X})] + \mathrm{E}_{q(\boldsymbol{\lambda})}[\log p(\boldsymbol{\lambda})] \\ & + \mathrm{E}_{q(\boldsymbol{\lambda})q(\boldsymbol{b},\mathbf{W})}[\log p(\boldsymbol{b}|\boldsymbol{\lambda})] + \mathrm{E}_{q(\boldsymbol{\Psi})}[\log p(\boldsymbol{\Psi})] \\ & + \mathrm{E}_{q(\boldsymbol{\Psi})q(\boldsymbol{b},\mathbf{W})}[\log p(\mathbf{W}|\boldsymbol{\Psi})] \\ & + \mathrm{E}_{q(\mathbf{Z})q(\boldsymbol{b},\mathbf{W})q(\mathbf{T})}[\log p(\mathbf{T}|\boldsymbol{b},\mathbf{W},\mathbf{Z})] \\ & + \mathrm{E}_{q(\mathbf{T})}[\log p(\boldsymbol{y}|\mathbf{T})] - \mathrm{E}_{q(\boldsymbol{\Phi})}[\log q(\boldsymbol{\Phi})] \\ & - \mathrm{E}_{q(\mathbf{Q})}[\log q(\mathbf{Q})] - \mathrm{E}_{q(\mathbf{Z})}[\log q(\mathbf{Z})] \\ & - \mathrm{E}_{q(\boldsymbol{\lambda})}[\log q(\boldsymbol{\lambda})] - \mathrm{E}_{q(\boldsymbol{\Psi})}[\log q(\boldsymbol{\Psi})] \\ & - \mathrm{E}_{q(\boldsymbol{b},\mathbf{W})}[\log q(\boldsymbol{b},\mathbf{W})] - \mathrm{E}_{q(\mathbf{T})}[\log q(\mathbf{T})]\end{aligned}$$

where the exponential form expectations of the distributions in the joint likelihood can be calculated as

$$\mathrm{E}_{q(\boldsymbol{\Phi})}[\log p(\boldsymbol{\Phi})] = \sum_{f=1}^{D}\sum_{s=1}^{R}\Bigg((\alpha_\phi - 1)\widetilde{\log \phi_s^f} - \frac{\widetilde{\phi_s^f}}{\beta_\phi} \\ - \log\Gamma(\alpha_\phi) - \alpha_\phi \log \beta_\phi\Bigg)$$

$$\mathrm{E}_{q(\boldsymbol{\Phi})q(\mathbf{Q})}[\log p(\mathbf{Q}|\boldsymbol{\Phi})] = \sum_{s=1}^{R}\Bigg(-\frac{1}{2}\operatorname{tr}(\operatorname{diag}(\widetilde{\boldsymbol{\phi}_s})\widetilde{\boldsymbol{q}_s\boldsymbol{q}_s^\top}) \\ - \frac{1}{2}D\log 2\pi + \frac{1}{2}\log|\operatorname{diag}(\widetilde{\boldsymbol{\phi}_s})|\Bigg)$$

$$\mathrm{E}_{q(\mathbf{Q})q(\mathbf{Z})}[\log p(\mathbf{Z}|\mathbf{Q},\mathbf{X})] = \sum_{i=1}^{N}\Bigg(-\frac{1}{2}\widetilde{\boldsymbol{z}_i^\top \boldsymbol{z}_i} + \boldsymbol{x}_i^\top \widetilde{\mathbf{Q}}\widetilde{\boldsymbol{z}_i} \\ - \frac{1}{2}\operatorname{tr}(\widetilde{\mathbf{Q}\mathbf{Q}^\top}\boldsymbol{x}_i\boldsymbol{x}_i^\top) - \frac{1}{2}R\log 2\pi\Bigg)$$

$$\mathrm{E}_{q(\boldsymbol{\lambda})}[\log p(\boldsymbol{\lambda})] = \sum_{o=1}^{L}\Bigg((\alpha_\lambda - 1)\widetilde{\log\lambda_o} - \frac{\widetilde{\lambda_o}}{\beta_\lambda} - \log\Gamma(\alpha_\lambda) \\ - \alpha_\lambda\log\beta_\lambda\Bigg)$$

$$\mathrm{E}_{q(\boldsymbol{\lambda})q(\boldsymbol{b},\mathbf{W})}[\log p(\boldsymbol{b}|\boldsymbol{\lambda})] = \sum_{o=1}^{L}\Bigg(-\frac{1}{2}\widetilde{\lambda_o}\widetilde{b_o^2} - \frac{1}{2}\log 2\pi \\ + \frac{1}{2}\log\widetilde{\lambda_o}\Bigg)$$

$$\mathrm{E}_{q(\boldsymbol{\Psi})}[\log p(\boldsymbol{\Psi})] = \sum_{s=1}^{R}\sum_{o=1}^{L}\Bigg((\alpha_\psi - 1)\widetilde{\log\psi_o^s} - \frac{\widetilde{\psi_o^s}}{\beta_\psi} \\ - \log\Gamma(\alpha_\psi) - \alpha_\psi\log\beta_\psi\Bigg)$$

$$\mathrm{E}_{q(\boldsymbol{\Psi})q(\boldsymbol{b},\mathbf{W})}[\log p(\mathbf{W}|\boldsymbol{\Psi})] = \\ \sum_{o=1}^{L}\Bigg(-\frac{1}{2}\operatorname{tr}(\operatorname{diag}(\widetilde{\boldsymbol{\psi}_o})\widetilde{\boldsymbol{w}_o\boldsymbol{w}_o^\top}) - \frac{1}{2}R\log 2\pi \\ + \frac{1}{2}\log|\operatorname{diag}(\widetilde{\boldsymbol{\psi}_o})|\Bigg)$$

$$\mathrm{E}_{q(\mathbf{Z})q(\boldsymbol{b},\mathbf{W})q(\mathbf{T})}[\log p(\mathbf{T}|\boldsymbol{b},\mathbf{W},\mathbf{Z})] = \sum_{o=1}^{L}\sum_{i=1}^{N}\Bigg(-\frac{1}{2}\widetilde{(t_i^o)^2} \\ + (\widetilde{\boldsymbol{w}_o^\top}\widetilde{\boldsymbol{z}_i} + \widetilde{b_o})\widetilde{t_i^o} - \frac{1}{2}\Big(\operatorname{tr}(\widetilde{\boldsymbol{w}_o\boldsymbol{w}_o^\top}\widetilde{\boldsymbol{z}_i\boldsymbol{z}_i^\top}) + 2\widetilde{b_o\boldsymbol{w}_o^\top}\widetilde{\boldsymbol{z}_i} + \widetilde{b_o^2}\Big) \\ - \frac{1}{2}\log 2\pi\Bigg)$$

$$\mathrm{E}_{q(\mathbf{T})}[\log p(\boldsymbol{y}|\mathbf{T})] = 0$$

and the negative entropies of the approximate posteriors in the ensemble are given as

$$\mathrm{E}_{q(\boldsymbol{\Phi})}[\log q(\boldsymbol{\Phi})] = \sum_{f=1}^{D}\sum_{s=1}^{R}(-\alpha(\phi_s^f) - \log\beta(\phi_s^f) \\ - \log\Gamma(\alpha(\phi_s^f)) - (1-\alpha(\phi_s^f))\psi(\alpha(\phi_s^f)))$$

$$\mathrm{E}_{q(\mathbf{Q})}[\log q(\mathbf{Q})] = \\ \sum_{s=1}^{R}\Bigg(-\frac{1}{2}D(\log 2\pi + 1) - \frac{1}{2}\log|\Sigma(\boldsymbol{q}_s)|\Bigg)$$

$$\mathrm{E}_{q(\mathbf{Z})}[\log q(\mathbf{Z})] = \sum_{i=1}^{N}\Bigg(-\frac{1}{2}R(\log 2\pi + 1) - \frac{1}{2}\log|\Sigma(\boldsymbol{z}_i)|\Bigg)$$

$$\mathrm{E}_{q(\boldsymbol{\lambda})}[\log q(\boldsymbol{\lambda})] = \sum_{o=1}^{L}(-\alpha(\lambda_o) - \log\beta(\lambda_o) - \log\Gamma(\alpha(\lambda_o)) - (1-\alpha(\lambda_o))\psi(\alpha(\lambda_o)))$$

$$\mathrm{E}_{q(\boldsymbol{\Psi})}[\log q(\boldsymbol{\Psi})] = \sum_{s=1}^{R}\sum_{o=1}^{L}(-\alpha(\psi_o^s) - \log\beta(\psi_o^s) - \log\Gamma(\alpha(\psi_o^s)) - (1-\alpha(\psi_o^s))\psi(\alpha(\psi_o^s)))$$

$$\mathrm{E}_{q(\boldsymbol{b},\mathbf{W})}[\log q(\boldsymbol{b},\mathbf{W})] = \sum_{o=1}^{L}\left(-\frac{1}{2}(R+1)(\log 2\pi + 1) - \frac{1}{2}\log|\Sigma(b_o, \boldsymbol{w}_o)|\right)$$

$$\mathrm{E}_{q(\mathbf{T})}[\log q(\mathbf{T})] = \sum_{o=1}^{L}\sum_{i=1}^{N}\left(-\frac{1}{2}(\log 2\pi + \Sigma(t_i^o)) - \log\mathcal{Z}_i^o\right)$$

where $\Gamma(\cdot)$ denotes the gamma function and $\psi(\cdot)$ denotes the digamma function. The only nonstandard distribution we need to operate on is the truncated normal distribution used for the auxiliary variables. From our model definition, the truncation points for each auxiliary variable are defined as

$$(l_i^o, u_i^o) = \begin{cases}(-\infty, 0) & \text{if } y_i^o = -1\\ (0, +\infty) & \text{otherwise}\end{cases}$$

where l_i^o and u_i^o denote the lower and upper truncation points, respectively. The normalization coefficient, the expectation, and the variance of the auxiliary variables can be calculated as

$$\begin{aligned}\mathcal{Z}_i^o &= \Phi(\beta_i^o) - \Phi(\alpha_i^o)\\ \widetilde{t_i^o} &= \widetilde{\boldsymbol{w}_o^\top}\widetilde{\boldsymbol{z}}_i + \widetilde{b_o} + \frac{\phi(\alpha_i^o) - \phi(\beta_i^o)}{\mathcal{Z}_i^o}\\ \widetilde{(t_i^o)^2} - \widetilde{t_i^o}^2 &= 1 + \frac{\alpha_i^o\phi(\alpha_i^o) - \beta_i^o\phi(\beta_i^o)}{\mathcal{Z}_i^o} - \frac{(\phi(\alpha_i^o) - \phi(\beta_i^o))^2}{(\mathcal{Z}_i^o)^2}\end{aligned}$$

where $\phi(\cdot)$ is the standardized normal probability density function and $\{\alpha_i^o, \beta_i^o\}$ are defined as

$$\begin{aligned}\alpha_i^o &= l_i^o - \widetilde{\boldsymbol{w}_o^\top}\widetilde{\boldsymbol{z}}_i - \widetilde{b_o}\\ \beta_i^o &= u_i^o - \widetilde{\boldsymbol{w}_o^\top}\widetilde{\boldsymbol{z}}_i - \widetilde{b_o}.\end{aligned}$$