Published in final edited form as:

Pattern Recognit Lett. 2014 March 1; 38: 132–141. doi:10.1016/j.patrec.2013.11.021.

Coupled dimensionality reduction and classification for supervised and semi-supervised multilabel learning

Mehmet Gönen*

Sage Bionetworks, Seattle, 98109 WA, USA

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 first introduce a novel Bayesian method that combines linear dimensionality reduction with linear binary classification for supervised multilabel learning and present a deterministic variational approximation algorithm to learn the proposed probabilistic model. We then extend the proposed method to find intrinsic dimensionality of the projected subspace using automatic relevance determination and to handle semi-supervised learning using a low-density assumption. We perform supervised learning experiments on four benchmark multilabel learning data sets by comparing our method with baseline linear dimensionality reduction algorithms. These experiments show that the proposed approach achieves good performance values in terms of hamming loss, average AUC, 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. We also show the effectiveness of our approach in finding intrinsic subspace dimensionality and semi-supervised learning tasks.

Keywords

Multilabel learning; Dimensionality reduction; Supervised learning; Semi-supervised learning; Variational approximation; Automatic relevance determination

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 (Tsoumakas et al., 2009). 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: (i) joint learning of the model parameters of distinct classifiers trained for each label (Boutell et al., 2004; Zhang and Zhou, 2007; Sun et al., 2008; Petterson and Caetano, 2010; Guo and Gu, 2011; Zhang, 2011; Zhang et al., 2011) and (ii) learning a shared subspace and doing classification in this subspace (Yu et al., 2005; Park and Lee, 2008; Ji and Ye, 2009; Rai and Daumé III, 2009; Ji

© 2013 Elsevier B.V. All rights reserved.

*Corresponding author. mehmet.gonen@sagebase.org (Mehmet Gönen).

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

et al., 2010; Wang et al., 2010; Zhang and Zhou, 2010). In this paper, we are focusing on the second approach.

Dimensionality reduction algorithms try to achieve two main goals: (i) removing the inherent noise to improve the prediction performance and (ii) obtaining low-dimensional visualizations for exploratory data analysis. *Principal component analysis* (PCA) (Pearson, 1901) and *linear discriminant analysis* (LDA) (Fisher, 1936) are two well-known algorithms for unsupervised and supervised 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, Yu et al. (2005) propose a supervised *latent semantic indexing* variant that makes use of multiple labels. Park and Lee (2008) and Wang et al. (2010) modify LDA algorithm for multilabel learning. Rai and Daumé III (2009) propose a probabilistic *canonical correlation analysis* method that can also be applied in semi-supervised settings. Ji et al. (2010) and Zhang and Zhou (2010) 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. Biem et al. (1997) propose a multilayer perceptron variant that performs coupled feature extraction and classification. Coupled training of the projection matrix and the classifier is also studied in the framework of support vector machines by introducing the projection matrix into the optimization problem solved (Chapelle et al., 2002; Pereira and Gordon, 2006). Gönen and Alpaydm (2010) introduce the same idea to a localized multiple kernel learning framework to capture multiple modalities that may exist in the data. There are also metric learning methods that try to transfer the neighborhood in the input space to the projected subspace in nearest neighbor settings (Goldberger et al., 2005; Globerson and Roweis, 2006; Weinberger and Saul, 2009). Sajama and Orlitsky (2005) use mixture models for each class to obtain better projections, whereas Mao et al. (2010) use 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. Yu et al. (2006) propose a supervised probabilistic PCA and an efficient solution method, but the algorithm is developed only for real outputs. Rish et al. (2008) formulate a supervised dimensionality reduction algorithm coupled with generalized linear models for binary classification and regression, and maximize a target function composed of input and output likelihood terms using an iterative algorithm.

In this paper, we propose novel supervised and semi-supervised multilabel learning methods where the linear projection matrix and the binary classification parameters 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 supervised multilabel learning called *Bayesian supervised multilabel learning* (BSML) and introduce a deterministic variational approximation for inference. Section 3 formulates our two variants: (i) BSML with *automatic relevance determination* (BSML+ARD) to find intrinsic dimensionality of the projected subspace and (ii) *Bayesian semi-supervised multilabel learning* (BSSML) to make use of unlabeled data. In Section 4, we discuss the key properties of our algorithms. Section 5 tests our algorithms on four benchmark multilabel data sets in different settings.

2. Coupled dimensionality reduction and classification for supervised multilabel learning

Performing dimensionality reduction and classification successively (with two different objective functions) may not result in a predictive subspace and may have low generalization performance. In order to find a better subspace, coupling dimensionality reduction and single-output supervised learning is previously proposed (Biem et al., 1997; Chapelle et al., 2002; Goldberger et al., 2005; Sajama and Orlitsky, 2005; Globerson and Roweis, 2006; Pereira and Gordon, 2006; Yu et al., 2006; Rish et al., 2008; Weinberger and Saul, 2009; Gönen and Alpaydm, 2010; Mao et al., 2010). We should consider the predictive performance of the target subspace while learning the projection matrix. In order to benefit from the correlation between the class labels in a multilabel learning scenario, 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 1 illustrates the probabilistic model for multilabel binary classification with a graphical model and its distributional assumptions. The data matrix $\mathbf{X}$ is used to project data points into a low-dimensional space using the projection matrix $\mathbf{Q}$. The low-dimensional representations of data points $\mathbf{Z}$ and the classification parameters $\{\mathbf{b}, \mathbf{W}\}$ are used to calculate the classification scores. Finally, the given class labels $\mathbf{Y}$ are generated from the auxiliary matrix $\mathbf{T}$, which is introduced to make the inference procedures efficient (Albert and Chib, 1993). We formulate a variational approximation procedure for inference in order to have a computationally efficient algorithm.

The notation we use throughout the manuscript is given in Table 1. The superscripts index the rows of matrices, whereas the subscripts index the columns of matrices and the entries of vectors. As short-hand notations, all priors in the model are denoted by $\Xi = \{\lambda, \Phi, \Psi\}$, where the remaining variables by $\Theta = \{\mathbf{b}, \mathbf{Q}, \mathbf{T}, \mathbf{W}, \mathbf{Z}\}$ and the hyper-parameters by $\zeta = \{\alpha_\lambda, \beta_\lambda, \alpha_\phi, \beta_\phi, \alpha_\psi, \beta_\psi\}$. Dependence on ζ is omitted for clarity throughout the manuscript. $N(\cdot; \mu, \Sigma)$ denotes the normal distribution with the mean vector μ and the covariance matrix Σ . $G(\cdot; \alpha, \beta)$ 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.

2.1. Inference using variational approximation

The variational methods use a lower bound on the marginal likelihood using an ensemble of factored posteriors to find the joint parameter distribution (Beal, 2003). 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

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

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

$$\begin{aligned}
q(\Phi) &= \prod_{f=1}^D \prod_{s=1}^R G(\phi_s^f; \alpha(\phi_s^f), \beta(\phi_s^f)) \\
q(\mathbf{Q}) &= \prod_{s=1}^R N(\mathbf{q}_s; \mu(\mathbf{q}_s), \Sigma(\mathbf{q}_s)) \\
q(\mathbf{Z}) &= \prod_{i=1}^N N(\mathbf{z}_i; \mu(\mathbf{z}_i), \Sigma(\mathbf{z}_i)) \\
q(\lambda) &= \prod_{o=1}^L G(\lambda_o; \alpha(\lambda_o), \beta(\lambda_o)) \\
q(\Psi) &= \prod_{o=1}^L \prod_{s=1}^R G(\psi_o^s; \alpha(\psi_o^s), \beta(\psi_o^s)) \\
q(\mathbf{b}, \mathbf{W}) &= \prod_{o=1}^L N\left(\begin{bmatrix} b_o \\ \mathbf{w}_o \end{bmatrix}; \mu(b_o, \mathbf{w}_o), \Sigma(b_o, \mathbf{w}_o)\right) \\
q(\mathbf{T}) &= \prod_{o=1}^L \prod_{i=1}^N TN(t_i^o; \mu(t_i^o), \Sigma(t_i^o), \rho(t_i^o))
\end{aligned}$$

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. $TN(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \rho(\cdot))$ denotes the truncated normal distribution with the mean vector $\boldsymbol{\mu}$, the covariance matrix $\boldsymbol{\Sigma}$, and the truncation rule $\rho(\cdot)$ such that $TN(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \rho(\cdot)) \propto N(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ if $\rho(\cdot)$ is true and $TN(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \rho(\cdot)) = 0$ otherwise.

We choose to model projected data instances explicitly (i.e., not marginalizing out them) and independently (i.e., assuming a distribution independent of other variables) in the factorable ensemble approximation in order to decouple the dimensionality reduction and classification parts. By doing this, we achieve to obtain update equations for $\mathbf{Q}$ and $\{\mathbf{b}, \mathbf{W}\}$ independent of each other.

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

$$\log p(\mathbf{Y}|\mathbf{X}) \geq \mathbb{E}_{q(\boldsymbol{\Theta}, \boldsymbol{\Xi})} [\log p(\mathbf{Y}, \boldsymbol{\Theta}, \boldsymbol{\Xi}|\mathbf{X})] - \mathbb{E}_{q(\boldsymbol{\Theta}, \boldsymbol{\Xi})} [\log q(\boldsymbol{\Theta}, \boldsymbol{\Xi})] \quad (1)$$

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

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

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

2.2. 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:

$$q(\Phi) = \prod_{f=1}^D \prod_{s=1}^R G\left(\phi_s^f; \alpha_\phi + 1/2, (1/\beta_\phi + \langle (q_s^f)^2 \rangle / 2)^{-1}\right) \quad (2)$$

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

$$q(\mathbf{Q}) = \prod_{s=1}^R N \left(\mathbf{q}_s; \sum (\mathbf{q}_s) \mathbf{X} \langle \mathbf{z}^s \rangle, (\text{diag}(\langle \phi_s \rangle) + \mathbf{X} \mathbf{X}^\top)^{-1} \right). \quad (3)$$

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

$$q(\mathbf{Z}) = \prod_{i=1}^N N \left(\mathbf{z}_i; \sum (\mathbf{z}_i) \left(\langle \mathbf{Q}^\top \rangle \mathbf{x}_i + \sum_{o=1}^L (\langle \mathbf{w}_o \rangle \langle t_i^o \rangle - \langle \mathbf{w}_o b_o \rangle) \right), \left(\mathbf{I} + \sum_{o=1}^L \langle \mathbf{w}_o \mathbf{w}_o^\top \rangle \right)^{-1} \right) \quad (4)$$

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:

$$q(\boldsymbol{\lambda}) = \prod_{o=1}^L G \left(\lambda_o; \alpha_\lambda + 1/2, (1/\beta_\lambda + \langle b_o^2 \rangle / 2)^{-1} \right) \quad (5)$$

$$q(\boldsymbol{\Psi}) = \prod_{o=1}^L \prod_{s=1}^R G \left(\psi_o^s; \alpha_\psi + 1/2, (1/\beta_\psi + \langle (w_o^s)^2 \rangle / 2)^{-1} \right). \quad (6)$$

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

$$q(\mathbf{b}, \mathbf{W}) = \prod_{o=1}^L N \left(\begin{bmatrix} b_o \\ \mathbf{w}_o \end{bmatrix}; \sum (b_o, \mathbf{w}_o) \begin{bmatrix} \mathbf{1}^\top \langle \mathbf{t}_o \rangle \\ \langle \mathbf{Z} \rangle \langle \mathbf{t}_o \rangle \end{bmatrix}, \begin{bmatrix} \langle \lambda_o \rangle + N & \mathbf{1}^\top \langle \mathbf{Z}^\top \rangle \\ \langle \mathbf{Z} \rangle \mathbf{1} & \text{diag}(\langle \psi_o \rangle) + \langle \mathbf{Z} \mathbf{Z}^\top \rangle \end{bmatrix}^{-1} \right). \quad (7)$$

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

$$q(\mathbf{T}) = \prod_{o=1}^L \prod_{i=1}^N TN(t_i^o; \langle \mathbf{w}_o^\top \rangle \langle \mathbf{z}_i \rangle + \langle b_o \rangle, 1, t_i^o y_i^o > 0) \quad (8)$$

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.

2.3. Complete algorithm

The complete inference algorithm is listed in Algorithm 1. The inference mechanism sequentially updates the approximate posterior distributions of the model parameters and the latent variables until convergence, which can be checked by monitoring the lower bound in (1). Exact form of the variational lower bound can be found in Appendix Appendix A. The first term of the lower bound corresponds to the sum of exponential forms 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. The posterior expectations needed for the algorithm can be found in Appendix Appendix B.

Algorithm 1 Bayesian Supervised Multilabel Learning

Require: $\mathbf{X}, \mathbf{Y}, R, \alpha_\lambda, \beta_\lambda, \alpha_\phi, \beta_\phi, \alpha_\psi$ and β_ψ

- 1 Initialize $q(\mathbf{Q}), q(\mathbf{Z}), q(\mathbf{b}, \mathbf{W})$, and $q(\mathbf{T})$ randomly
- 2 **repeat**
- 3 Update $q(\Phi)$ and $q(\mathbf{Q})$ using (2) and (3)
- 4 Update $q(\mathbf{Z})$ using (4)
- 5 Update $q(\lambda), q(\Psi)$, and $q(\mathbf{b}, \mathbf{W})$ using (5), (6), and (7)
- 6 Update $q(\mathbf{T})$ using (8)
- 7 **until** convergence $q(\mathbf{Q})$ and $q(\mathbf{b}, \mathbf{W})$

2.4. Prediction

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

$$p(\mathbf{z}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}) = \prod_{s=1}^R N(\mathbf{z}_*^s; \mu(\mathbf{q}_s)^\top \mathbf{x}_*, 1 + \mathbf{x}_*^\top \sum (\mathbf{q}_s) \mathbf{x}_*).$$

The predictive distribution of the auxiliary variable t_*^o can also be found by replacing $p(\mathbf{b}, \mathbf{W}|\mathbf{X}, \mathbf{Y})$ with its approximate posterior distribution $q(\mathbf{b}, \mathbf{W})$:

$$p(t_*^o|\mathbf{X}, \mathbf{Y}, \mathbf{z}_*) = N\left(t_*^o; \mu(b_o, \mathbf{w}_o)^\top \begin{bmatrix} 1 \\ \mathbf{z}_* \end{bmatrix}, [1 \ \mathbf{z}_*] \sum (b_o, \mathbf{w}_o) \begin{bmatrix} 1 \\ \mathbf{z}_* \end{bmatrix} + 1\right) \quad \forall o$$

and the predictive distribution of the class label y_*^o can be formulated using the auxiliary variable distribution:

$$p(y_*^o = +1|\mathbf{x}_*, \mathbf{X}, \mathbf{Y}) = \Phi(\mu(t_*^o) / \sum (t_*^o)) \quad \forall o$$

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

3. Extensions

This section introduces two variants derived from the base model we describe above.

3.1. Finding intrinsic subspace dimensionality using automatic relevance determination

The dimensionality of the projected subspace is generally selected using a cross-validation strategy or is fixed before learning. Instead of these two naive approaches, the intrinsic subspace dimensionality of the data can be identified while learning the model parameters. The typical choice is to use ARD (Neal, 1996) that defines independent multivariate Gaussian priors on the columns of the projection matrix.

The distributional assumptions for the column-wise prior case can be given as

$$\begin{aligned}\phi_s &\sim G(\phi_s; \alpha_\phi, \beta_\phi) \quad \forall s \\ q_s^f | \phi_s &\sim N(q_s^f; 0, \phi_s^{-1}) \quad \forall (f, s)\end{aligned}$$

and the approximate posterior distributions of the priors and the projection matrix become

$$\begin{aligned}q(\phi) &= \prod_{s=1}^R G(\phi_s; \alpha_\phi + D/2, (1/\beta_\phi + \langle \mathbf{q}_s^\top \mathbf{q}_s \rangle / 2)^{-1}) \\ q(\mathbf{Q}) &= \prod_{s=1}^R N(\mathbf{q}_s; \sum(\mathbf{q}_s) \mathbf{X} \langle \mathbf{z}^s \rangle, (\langle \phi_s \rangle \mathbf{I} + \mathbf{X} \mathbf{X}^\top)^{-1}).\end{aligned}$$

3.2. Coupled dimensionality reduction and classification for semi-supervised multilabel learning

Labeling large data collections may not be possible due to extensive labour required. In such cases, we should efficiently use a large number of unlabeled data points in addition to a few labeled data points (i.e., semi-supervised learning). Semi-supervised learning is not well-studied in the context of multilabel learning. There are a few attempts that formulate the problem as a matrix factorization problem (Liu et al., 2006; Chen et al., 2008). To our knowledge, Rai and Daumé III (2009), Qian and Davidson (2010), and Guo and Schuurmans (2012) are the only studies that consider dimensionality reduction and classification together for multilabel learning in a semi-supervised setup.

We modify our probabilistic model described above for semi-supervised learning assuming a low-density region between the classes (Lawrence and Jordan, 2005). We basically need to make the class labels partially observed and to introduce a new set of observed auxiliary variables denoted by $\mathbf{L}$. The distributional assumptions for $\mathbf{Y}$ and $\mathbf{L}$ are defined as follows:

$$\begin{aligned}y_i^o | t_i^o &\sim \begin{cases} \delta(y_i^o t_i^o > 1/2) & y_i^o \in \{\pm 1\} \\ 1 - \delta(1/2 \geq t_i^o \geq -1/2) & \text{otherwise} \end{cases} \quad \forall (o, i) \\ l_i^o | y_i^o &\sim \delta(y_i^o \in \{\pm 1\}) \quad \forall (o, i).\end{aligned}$$

The first distributional assumption has two main implications: (i) A low-density region is placed between the classes similar to the margin in support vector machines. (ii) Unlabeled data points are forced to be outside of this low-density region.

The approximate posterior distribution of the auxiliary variables can again be formulated as a product of truncated normal distributions:

$$\begin{aligned}q(\mathbf{T}) &= \prod_{o=1}^L \prod_{i \in L_o} TN(t_i^o; \langle \mathbf{w}_o^\top \rangle \langle \mathbf{z}_i \rangle + \langle b_o \rangle, 1, t_i^o y_i^o > 1/2) \\ &\quad \prod_{o=1}^L \prod_{i \in u_o} (\gamma_o^- \Phi(-1/2 - \langle \mathbf{w}_o^\top \rangle \langle \mathbf{z}_i \rangle - \langle b_o \rangle) TN(t_i^o; \langle \mathbf{w}_o^\top \rangle \langle \mathbf{z}_i \rangle + \langle b_o \rangle, 1, t_i^o < -1/2) \\ &\quad + \gamma_o^+ \Phi(\langle \mathbf{w}_o^\top \rangle \langle \mathbf{z}_i \rangle + \langle b_o \rangle - 1/2) TN(t_i^o; \langle \mathbf{w}_o^\top \rangle \langle \mathbf{z}_i \rangle + \langle b_o \rangle, 1, t_i^o > 1/2)) (Z_i^o)^{-1}\end{aligned}$$

where $L_o = \{i: y_i^o \in \{\pm 1\}\}$, $u_o = \{i: y_i^o \notin \{\pm 1\}\}$, γ_o^- is the prior probability of the negative class for label o , γ_o^+ is the prior probability of the positive class for label o , and Z_i^o is the normalization coefficient calculated for the unlabeled data point $\mathbf{x}_i$ and label o . The predictive distribution of the class label y_{*}^o can be found as

$$p(y_*^o = +1 | \mathbf{x}_*, \mathbf{X}, \mathbf{Y}) = (Z_*^o)^{-1} \Phi((\mu(t_*^o) - 1/2) / \sum (t_*^o)) \quad \forall o$$

where Z_*^o is the normalization coefficient calculated for the test data point $\mathbf{x}_*$ and label o .

4. Discussion

The most time-consuming updates of our algorithm are the covariance calculations of (3), (4), and (7) whose time complexities can be given as $O(RD^3)$, $O(R^3)$, and $O(LR^3)$, respectively. If the number of output labels is not very large compared to the input space dimensionality, updating the projection matrix $\mathbf{Q}$ using (3) is the most time-consuming step, which requires inverting R different $D \times D$ 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. Note that having a large number of training instances does not affect the running time heavily due to simple update rules and posterior expectation calculations for the auxiliary matrix $\mathbf{T}$.

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 the number of output labels is larger than the projected subspace dimensionality (i.e., $L > R$), 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 (Tso, 1981).

We modify the precision priors for the projection matrix to determine the dimensionality of the projected subspace automatically. These priors can also be modified to decide which features should be used by changing entry-wise priors on the projection matrix with row-wise sparse priors. This formulation allows us to perform feature selection and multilabel classification at the same time.

5. Experiments

We use four widely used benchmark data sets, namely, Emotions, Scene, TMC2007, and Yeast, from different domains to compare our algorithms with the baseline algorithms using provided train/test splits. These data sets are publicly available from <http://mulan.sourceforge.net/datasets.html> and their characteristics can be summarized as:

- Emotions: $N_{\text{train}} = 391$, $N_{\text{test}} = 202$, $D = 72$, and $L = 6$,
- Scene: $N_{\text{train}} = 1211$, $N_{\text{test}} = 1196$, $D = 294$, and $L = 6$,
- TMC2007: $N_{\text{train}} = 21519$, $N_{\text{test}} = 7077$, $D = 500$, and $L = 22$,
- Yeast: $N_{\text{train}} = 1500$, $N_{\text{test}} = 917$, $D = 103$, and $L = 14$.

Four popular performance measures for multilabel learning, namely, *hamming loss*, *average Area Under the Curve (AUC)*, *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. Average AUC is the average area under the Receiver Operating Characteristic (ROC) curve over the labels. The larger the value of average AUC, 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. We also report training times for all of the algorithms to compare their computational requirements.

5.1. Supervised learning 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 (Pearson, 1901), *multilabel dimensionality reduction via dependency maximization* (MDDM) (Zhang and Zhou, 2010), *multilabel least squares* (MLLS) (Ji et al., 2010), and *multilabel linear discriminant analysis* (MLDA) (Wang et al., 2010). 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. 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.

Figure 2 gives the classification results on Emotions data set. We perform experiments with $R = 1, 2, \dots, 6$ for all of the methods except MLDA and with $R = 1, 2, \dots, 5$ 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 average AUC, 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. When we compare training times, we see that all baseline algorithms are trained in two seconds, whereas PROBIT requires three seconds and BSML takes three to five seconds depending on the subspace dimensionality.

The classification results on Scene data set are given in Figure 3. We perform experiments with $R = 1, 2, \dots, 6$ for all of the methods except MLDA and with $R = 1, 2, \dots, 5$ 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 and average AUC 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 . All baseline algorithms take less than one minute to train, whereas BSML requires more than one minute only when $R = 6$.

Figure 4 shows the classification results on TMC2007 data set. We perform experiments with $R = 1, 2, \dots, 22$ for all of the methods except MLDA and with $R = 1, 2, \dots, 21$ for MLDA. We see that BSML clearly outperforms all of the baseline dimensionality reduction algorithms in terms of hamming loss, average AUC (only when $R > 1$), macro F_1 (only

when $R < 13$), and micro F_1 . However, the performance values are not as good as PROBIT due to the large number of class labels (i.e., $L = 22$). The performance difference between PROBIT and BSML in terms of hamming loss is around two per cent with only two dimensions. All of the baseline dimensionality reduction algorithms take less than ten minutes to train, whereas PROBIT needs around 25 minutes and BSML requires around one hour when $R = 22$. This shows that our Bayesian formulation can be scaled up to large data sets with tens of thousands of training instances.

The classification results on Yeast data set are shown in Figure 5. We perform experiments with $R = 1, 2, \dots, 14$ for all of the methods except MLDA and with $R = 1, 2, \dots, 13$ for MLDA. 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. BSML is the best algorithm in terms of average AUC for $R > 2$ and it is the only algorithm that obtains consistently better average AUC values than PROBIT. When the projected subspace is two- or three-dimensional, BSML is clearly better than all of the baseline dimensionality reduction algorithms in terms of macro F_1 and micro F_1 . We also see that all of the algorithms can be trained under one minute on Yeast data set.

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 (not reported here) 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. Figure 6 shows two-dimensional embeddings of training data points and classification boundaries obtained by MLDA and BSML on Emotions data set. The 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 quiet-still and sad-lonely, and the negative correlation between relaxing-calm and angry-fearful.

5.2. Automatic relevance determination experiments

We perform another set of supervised learning experiments on four benchmark data sets to test whether our method can find the intrinsic subspace dimensionality automatically. We also implement BSML+ARD in Matlab and take 500 iterations for both methods. The default hyper-parameter values for BSML and BSML+ARD are selected as $(\alpha_\lambda, \beta_\lambda, \alpha_\phi, \beta_\phi, \alpha_\psi, \beta_\psi) = (1, 1, 1, 1, 1, 1)$ and $(\alpha_\lambda, \beta_\lambda, \alpha_\phi, \beta_\phi, \alpha_\psi, \beta_\psi) = (1, 1, 10^{-10}, 10^{+10}, 1, 1)$, respectively. For each dataset, we set the subspace dimensionality to the number of output labels (i.e., $R = L$).

The performances of BSML and BSML+ARD on benchmark data sets in terms of hamming loss are as follows:

- Emotions: BSML = 0.2153 and BSML+ARD = 0.2186,
- Scene: BSML = 0.1150 and BSML+ARD = 0.1356,
- TMC2007: BSML = 0.0578 and BSML+ARD = 0.0609,
- Yeast: BSML = 0.2053 and BSML+ARD = 0.2033.

We can see that BSML+ARD achieves comparable hamming loss values with respect to BSML and successfully eliminate some of the dimensions in order to find the intrinsic dimensionality. Two- or three-dimensional subspaces are enough to obtain a good prediction performance on Emotions (uses two dimensions), Scene (uses three dimensions), and Yeast (uses three dimensions) data sets, whereas TMC2007 data set requires eight dimensions.

5.3. Semi-supervised learning experiments

In order to test the performance of BSSML, we perform semi-supervised learning experiments on each data set using the transductive learning setup. We implement BSSML in Matlab and take 500 iterations. The default hyper-parameter values for BSSML are selected as $(\alpha_\lambda, \beta_\lambda, \alpha_\phi, \beta_\phi, \alpha_\psi, \beta_\psi) = (1, 1, 1, 1, 1, 1)$. We set γ^- and γ^+ to 0.5 not to impose any prior information on the unlabeled data points. We compare our new algorithm BSSML with *semi-supervised dimension reduction for multilabel classification* (SSDRMC) (Qian and Davidson, 2010). We also implement SSDRMC in Matlab and use the provided default parameter values.

The performances of BSSML and SSDRMC on benchmark data sets in terms of hamming loss are as follows:

- Emotions: SSDRMC = 0.2805 and BSSML = 0.2021,
- Scene: SSDRMC = 0.1345 and BSSML = 0.1164,
- TMC2007: SSDRMC = NA and BSSML = 0.0590,
- Yeast: SSDRMC = 0.2485 and BSSML = 0.2035.

Note that we could not report SSDRMC result on TMC2007 data set due to excessive computational need. We can see that, on Emotions, Scene, and Yeast data sets, BSSML is significantly better than SSDRMC in terms of hamming loss.

6. Conclusions

We present a Bayesian supervised multilabel learning method that couples linear dimensionality reduction and linear binary classification. We then provide detailed derivations for supervised learning using a deterministic variational approximation approach. We also formulate two variants: (i) an automatic relevance determination variant to find intrinsic dimensionality of the projected subspace and (ii) a semi-supervised learning variant with a low-density region between the classes to make use of unlabeled data. Matlab implementations of our algorithms BSML and BSSML are publicly available at <http://users.ics.aalto.fi/gonen/bssml/>.

Supervised learning experiments 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. Automatic relevance determination and semi-supervised learning experiments also show the effectiveness of our formulation in different settings.

The proposed models can be extended in different directions: First, we can use a nonlinear dimensionality reduction step before multilabel classification step using kernels instead of data matrix. Second, we can use a nonlinear classification algorithm such as Gaussian processes instead of probit model in our formulation to increase the prediction performance. 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).

Acknowledgments

Most of this work has been done while the author was working at the Helsinki Institute for Information Technology HIIT, Department of Information and Computer Science, Aalto University, Espoo, Finland. This work was financially supported by the Integrative Cancer Biology Program of the National Cancer Institute (grant no 1U54CA149237) and the Academy of Finland (Finnish Centre of Excellence in Computational Inference Research COIN, grant no 251170). A preliminary version of this work appears in Gönen (2012).

Appendix A. Variational lower bound

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

$$L = E_{q(\Theta, \Xi)}[\log p(\mathbf{Y}, \Theta, \Xi | \mathbf{X})] - E_{q(\Theta, \Xi)}[\log q(\Theta, \Xi)]$$

where the joint likelihood is defined as

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

Using these definitions, the variational lower bound becomes

$$\begin{aligned} L = & E_{q(\Phi)}[\log p(\Phi)] + E_{q(\Phi)q(\mathbf{Q})}[\log p(\mathbf{Q}|\Phi)] + E_{q(\mathbf{Q})q(\mathbf{Z})}[\log p(\mathbf{Z}|\mathbf{Q}, \mathbf{X})] + E_{q(\lambda)}[\log p(\lambda)] \\ & + E_{q(\lambda)q(\mathbf{b}, \mathbf{W})}[\log p(\mathbf{b}|\lambda)] + E_{q(\Psi)}[\log p(\Psi)] + E_{q(\Psi)q(\mathbf{b}, \mathbf{W})}[\log p(\mathbf{W}|\Psi)] \\ & + E_{q(\mathbf{Z})q(\mathbf{b}, \mathbf{W})q(\mathbf{T})}[\log p(\mathbf{T}|\mathbf{b}, \mathbf{W}, \mathbf{Z})] + E_{q(\mathbf{T})}[\log p(\mathbf{Y}|\mathbf{T})] - E_{q(\Phi)}[\log q(\Phi)] - E_{q(\mathbf{Q})}[\log q(\mathbf{Q})] \\ & - E_{q(\mathbf{Z})}[\log q(\mathbf{Z})] - E_{q(\lambda)}[\log q(\lambda)] - E_{q(\Psi)}[\log q(\Psi)] - E_{q(\mathbf{b}, \mathbf{W})}[\log q(\mathbf{b}, \mathbf{W})] - E_{q(\mathbf{T})}[\log q(\mathbf{T})] \end{aligned}$$

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

$$\begin{aligned} E_{q(\Phi)}[\log p(\Phi)] &= \sum_{f=1}^D \sum_{s=1}^R ((\alpha_\phi - 1) \langle \log \phi_s^f \rangle - \langle \phi_s^f \rangle / \beta_\phi - \log \Gamma(\alpha_\phi) - \alpha_\phi \log \beta_\phi) \\ E_{q(\Phi)q(\mathbf{Q})}[\log p(\mathbf{Q}|\Phi)] &= \sum_{s=1}^R (-\text{tr}(\text{diag}(\langle \phi_s \rangle) \langle \mathbf{q}_s \mathbf{q}_s^\top \rangle) / 2 - D \log 2\pi / 2 + \log |\text{diag}(\langle \phi_s \rangle)| / 2) \\ E_{q(\mathbf{Q})q(\mathbf{Z})}[\log p(\mathbf{Z}|\mathbf{Q}, \mathbf{X})] &= \sum_{i=1}^N (-\langle \mathbf{z}_i^\top \mathbf{z}_i \rangle / 2 + \langle \mathbf{z}_i^\top \rangle \langle \mathbf{Q}^\top \rangle \mathbf{x}_i - \text{tr}(\langle \mathbf{Q} \mathbf{Q}^\top \rangle \mathbf{x}_i \mathbf{x}_i^\top) / 2 - R \log 2\pi / 2) \\ E_{q(\lambda)}[\log p(\lambda)] &= \sum_{o=1}^L ((\alpha_\lambda - 1) \langle \log \lambda_o \rangle - \langle \lambda_o \rangle / \beta_\lambda - \log \Gamma(\alpha_\lambda) - \alpha_\lambda \log \beta_\lambda) \\ E_{q(\lambda)q(\mathbf{b}, \mathbf{W})}[\log p(\mathbf{b}|\lambda)] &= \sum_{o=1}^L (-\langle \lambda_o \rangle \langle b_o^2 \rangle / 2 - \log 2\pi / 2 + \log \langle \lambda_o \rangle / 2) \\ E_{q(\Psi)}[\log p(\Psi)] &= \sum_{s=1}^R \sum_{o=1}^L ((\alpha_\psi - 1) \langle \log \psi_o^s \rangle - \langle \psi_o^s \rangle / \beta_\psi - \log \Gamma(\alpha_\psi) - \alpha_\psi \log \beta_\psi) \\ E_{q(\Psi)q(\mathbf{b}, \mathbf{W})}[\log p(\mathbf{W}|\Psi)] &= \sum_{o=1}^L (-\text{tr}(\text{diag}(\langle \psi_o \rangle) \langle \mathbf{w}_o \mathbf{w}_o^\top \rangle) / 2 - R \log 2\pi / 2 + \log |\text{diag}(\langle \psi_o \rangle)| / 2) \\ E_{q(\mathbf{Z})q(\mathbf{b}, \mathbf{W})q(\mathbf{T})}[\log p(\mathbf{T}|\mathbf{b}, \mathbf{W}, \mathbf{Z})] &= \sum_{o=1}^L \sum_{i=1}^N (-\langle (t_i^o)^2 \rangle / 2 + \langle t_i^o \rangle (\langle \mathbf{w}_o^\top \rangle \langle \mathbf{z}_i \rangle + \langle b_o \rangle) \\ &\quad - (\text{tr}(\langle \mathbf{w}_o \mathbf{w}_o^\top \rangle \langle \mathbf{z}_i \mathbf{z}_i^\top \rangle) + 2 \langle \mathbf{z}_i^\top \rangle \langle \mathbf{w}_o b_o \rangle + \langle b_o^2 \rangle) / 2 - \log 2\pi / 2) \\ E_{q(\mathbf{T})}[\log p(\mathbf{Y}|\mathbf{T})] &= 0 \end{aligned}$$

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

$$\begin{aligned}
 E_{q(\Phi)}[\log q(\Phi)] &= \sum_{f=1}^D \sum_{s=1}^R (-\alpha(\phi_s^f) - \log \beta(\phi_s^f) - \log \Gamma(\alpha(\phi_s^f)) - (1 - \alpha(\phi_s^f))\psi(\alpha(\phi_s^f))) \\
 E_{q(\mathbf{Q})}[\log q(\mathbf{Q})] &= \sum_{s=1}^R (-D(\log 2\pi + 1)/2 - \log |\sum(\mathbf{q}_s)|/2) \\
 E_{q(\mathbf{Z})}[\log q(\mathbf{Z})] &= \sum_{i=1}^N (-R(\log 2\pi + 1)/2 - \log |\sum(\mathbf{z}_i)|/2) \\
 E_{q(\lambda)}[\log q(\lambda)] &= \sum_{o=1}^L (-\alpha(\lambda_o) - \log \beta(\lambda_o) - \log \Gamma(\alpha(\lambda_o)) - (1 - \alpha(\lambda_o))\psi(\alpha(\lambda_o))) \\
 E_{q(\Psi)}[\log q(\Psi)] &= \sum_{s=1}^R \sum_{o=1}^L (-\alpha(\psi_o^s) - \log \beta(\psi_o^s) - \log \Gamma(\alpha(\psi_o^s)) - (1 - \alpha(\psi_o^s))\psi(\alpha(\psi_o^s))) \\
 E_{q(\mathbf{b}, \mathbf{W})}[\log q(\mathbf{b}, \mathbf{W})] &= \sum_{o=1}^L (-(R+1)(\log 2\pi + 1)/2 - \log |\sum(b_o, \mathbf{w}_o)|/2) \\
 E_{q(\mathbf{T})}[\log q(\mathbf{T})] &= \sum_{o=1}^L \sum_{i=1}^N (-(\log 2\pi + \sum(t_i^o))/2 - \log Z_i^o)
 \end{aligned}$$

where $\Gamma(\cdot)$ denotes the gamma function and $\psi(\cdot)$ denotes the digamma function.

Appendix B. Posterior expectations

The posterior expectations needed in order to update approximate posterior distributions and to calculate the lower bound can be given as

$$\begin{aligned}
\langle \phi_s^f \rangle &= \alpha(\phi_s^f) \beta(\phi_s^f) & \forall(f, s) \\
\langle \log \phi_s^f \rangle &= \psi(\alpha(\phi_s^f)) + \log \beta(\phi_s^f) & \forall(f, s) \\
\langle \phi_s \rangle &= \left[\alpha(\phi_s^1) \beta(\phi_s^1) \quad \alpha(\phi_s^2) \beta(\phi_s^2) \quad \cdots \quad \alpha(\phi_s^D) \beta(\phi_s^D) \right]^\top & \forall s \\
\langle (q_s^f)^2 \rangle &= \mu(q_s^f)^2 + \sum(q_s^f) & \forall(f, s) \\
\langle \mathbf{q}_s \rangle &= \left[\mu(q_s^1) \quad \mu(q_s^2) \quad \cdots \quad \mu(q_s^D) \right]^\top & \forall s \\
\langle \mathbf{q}_s \mathbf{q}_s^\top \rangle &= \mu(\mathbf{q}_s) \mu(\mathbf{q}_s)^\top + \sum(\mathbf{q}_s) & \forall s \\
\langle \mathbf{Q} \rangle &= [\mu(\mathbf{q}_1) \quad \mu(\mathbf{q}_2) \quad \cdots \quad \mu(\mathbf{q}_R)] \\
\langle \mathbf{Q} \mathbf{Q}^\top \rangle &= \sum_{s=1}^R (\mu(\mathbf{q}_s) \mu(\mathbf{q}_s)^\top + \sum(\mathbf{q}_s)) \\
\langle \mathbf{z}^s \rangle &= \left[\mu(z_1^s) \quad \mu(z_2^s) \quad \cdots \quad \mu(z_N^s) \right]^\top & \forall s \\
\langle \mathbf{z}_i \rangle &= \left[\mu(z_i^1) \quad \mu(z_i^2) \quad \cdots \quad \mu(z_i^R) \right]^\top & \forall i \\
\langle \mathbf{z}_i^\top \mathbf{z}_i \rangle &= \sum_{s=1}^R (\mu(z_i^s)^2 + \sum(z_i^s)) & \forall i \\
\langle \mathbf{z}_i \mathbf{z}_i^\top \rangle &= \mu(\mathbf{z}_i) \mu(\mathbf{z}_i)^\top + \sum(\mathbf{z}_i) & \forall s \\
\langle \mathbf{Z} \rangle &= [\mu(\mathbf{z}_1) \quad \mu(\mathbf{z}_2) \quad \cdots \quad \mu(\mathbf{z}_N)] \\
\langle \mathbf{Z} \mathbf{Z}^\top \rangle &= \sum_{i=1}^N (\mu(\mathbf{z}_i) \mu(\mathbf{z}_i)^\top + \sum(\mathbf{z}_i)) \\
\langle \lambda_o \rangle &= \alpha(\lambda_o) \beta(\lambda_o) & \forall o \\
\langle \log \lambda_o \rangle &= \psi(\alpha(\lambda_o)) + \log \beta(\lambda_o) & \forall o \\
\langle \psi_o^s \rangle &= \alpha(\psi_o^s) \beta(\psi_o^s) & \forall(s, o) \\
\langle \log \psi_o^s \rangle &= \psi(\alpha(\psi_o^s)) + \log \beta(\psi_o^s) & \forall(s, o) \\
\langle \psi_o \rangle &= \left[\alpha(\psi_o^1) \beta(\psi_o^1) \quad \alpha(\psi_o^2) \beta(\psi_o^2) \quad \cdots \quad \alpha(\psi_o^R) \beta(\psi_o^R) \right]^\top & \forall o \\
\langle b_o \rangle &= \mu(b_o) & \forall o \\
\langle b_o^2 \rangle &= \mu(b_o)^2 + \sum(b_o) & \forall o \\
\langle (w_o^s)^2 \rangle &= \mu(w_o^s)^2 + \sum(w_o^s) & \forall(s, o) \\
\langle \mathbf{w}_o \rangle &= \left[\mu(w_o^1) \quad \mu(w_o^2) \quad \cdots \quad \mu(w_o^R) \right]^\top & \forall o \\
\langle \mathbf{w}_o \mathbf{w}_o^\top \rangle &= \mu(\mathbf{w}_o) \mu(\mathbf{w}_o)^\top + \sum(\mathbf{w}_o) & \forall o.
\end{aligned}$$

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

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

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

$$\begin{aligned}
Z_i^o &= \Phi(\beta_i^o) - \Phi(\alpha_i^o) & \forall(o, i) \\
\langle t_i^o \rangle &= \langle \mathbf{w}_o^\top \rangle \langle \mathbf{z}_i \rangle + \langle b_o \rangle + (\phi(\alpha_i^o) - \phi(\beta_i^o)) / Z_i^o & \forall(o, i) \\
\langle (t_i^o)^2 \rangle - (\langle t_i^o \rangle)^2 &= 1 + (\alpha_i^o \phi(\alpha_i^o) - \beta_i^o \phi(\beta_i^o)) / Z_i^o - (\phi(\alpha_i^o) - \phi(\beta_i^o))^2 / (Z_i^o)^2 & \forall(o, i)
\end{aligned}$$

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

$$\begin{aligned}\alpha_i^o &= l_i^o - \langle \mathbf{w}_o^\top \rangle \langle \mathbf{z}_i \rangle - \langle b_o \rangle \quad \forall(o, i) \\ \beta_i^o &= u_i^o - \langle \mathbf{w}_o^\top \rangle \langle \mathbf{z}_i \rangle - \langle b_o \rangle \quad \forall(o, i).\end{aligned}$$

References

- Albert JH, Chib S. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*. 1993; 88:669–679.
- Beal, MJ. Ph D thesis. The Gatsby Computational Neuroscience Unit; University College London: 2003. Variational algorithms for approximate Bayesian inference.
- Biem A, Katagiri S, Juang B-H. Pattern recognition using discriminative feature extraction. *IEEE Transactions on Signal Processing*. 1997; 45:500–504.
- Boutell MR, Luo J, Shen X, Brown CM. Learning multi-label scene classification. *Pattern Recognition*. 2004; 37:1757–1771.
- Chapelle O, Vapnik V, Bousquet O, Mukherjee S. Choosing multiple parameters for support vector machines. *Machine Learning*. 2002; 46:131–159.
- Chen, G.; Song, Y.; Wang, F.; Zhang, C. Semi-supervised multi-label learning by solving a Sylvester equation. *Proceedings of the 8th SIAM International Conference on Data Mining*; 2008.
- Fisher RA. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*. 1936; 7(Part II):179–188.
- Globerson A, Roweis S. Metric learning by collapsing classes. *Advances in Neural Information Processing Systems*. 2006; 18
- Goldberger J, Roweis S, Hinton G, Salakhutdinov R. Neighbourhood components analysis. *Advances in Neural Information Processing Systems*. 2005; 17
- Gönen, M. Bayesian supervised multilabel learning with coupled embedding and classification. *Proceedings of the 12th SIAM International Conference on Data Mining*; 2012.
- Gönen M, Alpaydin E. Supervised learning of local projection kernels. *Neurocomputing*. 2010; 73:1694–1703.
- Guo, Y.; Gu, S. Multi-label classification using conditional dependency networks. *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*; 2011.
- Guo, Y.; Schuurmans, D. Semi-supervised multi-label classification: A simultaneous large-margin, subspace learning approach. *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*; 2012.
- Ji S, Tang L, Yu S, Ye J. A shared-subspace learning framework for multi-label classification. *ACM Transactions on Knowledge Discovery from Data*. 2010; 4(2):8:1–8:29.
- Ji, S.; Ye, J. Linear dimensionality reduction for multi-label classification. *Proceedings of the 21st International Joint Conference on Artificial Intelligence*; 2009.
- Lawrence ND, Jordan MI. Semi-supervised learning via Gaussian processes. *Advances in Neural Information Processing Systems*. 2005; 17
- Liu, Y.; Jin, R.; Yang, L. Semi-supervised multi-label learning by constrained non-negative matrix factorization. *Proceedings of the 21st AAAI Conference on Artificial Intelligence*; 2006.
- Mao, K.; Liang, F.; Mukherjee, S. Supervised dimension reduction using Bayesian mixture modeling. *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*; 2010.
- Neal, RM. *Lecture Notes in Statistics*. Vol. 118. Springer; New York, NY: 1996. *Bayesian Learning for Neural Networks*.
- Park CH, Lee M. On applying linear discriminant analysis for multi-labeled problems. *Pattern Recognition Letters*. 2008; 29:878–887.
- Pearson K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*. 1901; 2:559–572.
- Pereira, F.; Gordon, G. The support vector decomposition machine. *Proceedings of the 23rd International Conference on Machine Learning*; 2006.

- Petterson J, Caetano T. Reverse multi-label learning. *Advances in Neural Information Processing Systems*. 2010; 23
- Qian, B.; Davidson, I. Semi-supervised dimension reduction for multi-label classification. *Proceedings of the 24th AAAI Conference on Artificial Intelligence*; 2010.
- Rai P, Daumé H III. Multi-label prediction via sparse infinite CCA. *Advances in Neural Information Processing Systems*. 2009; 22
- Rish, I.; Grabarnik, G.; Cecchi, G.; Pereira, F.; Gordon, GJ. Closed-form supervised dimensionality reduction with generalized linear models. *Proceedings of the 25th International Conference on Machine Learning*; 2008.
- Sajama; Orlitsky, A. Supervised dimensionality reduction using mixture models. *Proceedings of the 22nd International Conference on Machine Learning*; 2005.
- Sun, L.; Ji, S.; Ye, J. Hypergraph spectral learning for multi-label classification. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; 2008.
- Tso MK-S. Reduced-rank regression and canonical analysis. *Journal of the Royal Statistical Society: Series B (Methodological)*. 1981; 43:183–189.
- Tsoumakas, G.; Katakis, I.; Vlahavas, I. Mining multi-label data. In: Maimon, O.; Rokach, L., editors. *Data Mining and Knowledge Discovery Handbook*. Springer; 2009. p. 667-685.
- Wang, H.; Ding, C.; Huang, H. Multi-label linear discriminant analysis. *Proceedings of the 11th European Conference on Computer Vision*; 2010.
- Weinberger KQ, Saul LK. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*. 2009; 10:207–244.
- Yu, K.; Yu, S.; Tresp, V. Multi-label informed latent semantic indexing. *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*; 2005.
- Yu, S.; Yu, K.; Tresp, V.; Kriegel, H-P.; Wu, M. Supervised probabilistic principal component analysis. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; 2006.
- Zhang, M-L. LIFT: Multi-label learning with label-specific features. *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*; 2011.
- Zhang M-L, Zhou Z-H. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*. 2007; 40:2038–2048.
- Zhang, W.; Xue, X.; Fan, J.; Huang, X.; Wu, B.; Liu, M. Multi-kernel multi-label learning with max-margin concept network. *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*; 2011.
- Zhang Y, Zhou Z-H. Multilabel dimensionality reduction via dependence maximization. *ACM Transactions on Knowledge Discovery from Data*. 2010; 4(3):14:1–14:21.

- * Introduces a novel Bayesian multilabel learning algorithm
- * Combines dimensionality reduction and classification for (semi-)supervised setup
- * Uses a deterministic variational approximation for efficient inference
- * Achieves good performance values in terms of hamming loss, average AUC, macro F1, and micro F1
- * Obtains very useful low-dimensional embeddings for exploratory data analysis

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

Figure 1.
Coupled dimensionality reduction and classification for supervised multilabel learning.

Figure 2.
Comparison of algorithms on Emotions data set.

Figure 3.
Comparison of algorithms on Scene data set.

Figure 4.
Comparison of algorithms on TMC2007 data set.

Figure 5.
Comparison of algorithms on Yeast data set.

Figure 6. Two-dimensional embeddings obtained by (a) MLDA and (b) BSML on Emotions data set.

Table 1

List of notation.

D	Input space dimensionality
N	Number of training instances
R	Projected subspace dimensionality
L	Number of output labels
$\mathbf{X} \in \mathbb{R}^{D \times N}$	Data matrix (i.e., collection of training instances)
$\mathbf{Q} \in \mathbb{R}^{D \times R}$	Projection matrix (i.e., dimensionality reduction parameters)
$\Phi \in \mathbb{R}^{D \times R}$	Priors for projection matrix (i.e., precision priors)
$\mathbf{Z} \in \mathbb{R}^{R \times N}$	Projected data matrix (i.e., low-dimensional representation of instances)
$\mathbf{W} \in \mathbb{R}^{L \times R}$	Weight matrix (i.e., classification parameters for labels)
$\Psi \in \mathbb{R}^{L \times R}$	Priors for weight matrix (i.e., precision priors)
$\mathbf{b} \in \mathbb{R}^L$	Bias vector (i.e., bias parameters for labels)
$\lambda \in \mathbb{R}^L$	Priors for bias vector (i.e., precision priors)
$\mathbf{T} \in \mathbb{R}^{L \times N}$	Auxiliary matrix (i.e., discriminant outputs)
$\mathbf{Y} \in \{\pm 1\}^{L \times N}$	Label matrix (i.e., true labels of training instances)