Multi-Label Dimensionality Reduction via Dependence Maximization

YIN ZHANG and ZHI-HUA ZHOU
Nanjing University, China

Multi-label learning deals with data associated with multiple labels simultaneously. Like other data mining and machine learning tasks, multi-label learning also suffers from the *curse of dimensionality*. Dimensionality reduction has been studied for many years, however, multi-label dimensionality reduction remains almost untouched. In this paper, we propose a multi-label dimensionality reduction method, MDDM, with two kinds of projection strategies, attempting to project the original data into a lower-dimensional feature space maximizing the dependence between the original feature description and the associated class labels. Based on the Hilbert-Schmidt Independence Criterion, we derive a eigen-decomposition problem which enables the dimensionality reduction process to be efficient. Experiments validate the performance of MDDM.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Data Mining; I.2.6 [**Artificial Intelligence**]: Learning

General Terms: Algorithms, Design, Experimentation

Additional Key Words and Phrases: Dimensionality reduction, multi-label learning

1. INTRODUCTION

In traditional supervised learning, each instance is associated with *one* label that indicates its concept class belongingness. In many real-world problems, however, one object usually inheres multiple concepts simultaneously. For example, in text categorization, a document on national health service belongs to several predefined topics such as *government* and *health* simultaneously; in image or video annotation, an image showing a tiger in woods is associated with several annotated words such as *tiger* and *trees* simultaneously. One label per instance is out of its capability to describe such scenario, and therefore *multi-label learning* has thus attracted much attention. Under the framework of multi-label learning, each instance is associated with multiple labels, indicating the concepts it belongs to. Multi-label learning has already been applied to web page classification [Ueda and Saito 2003; Kazawa et al. 2005; Zhang and Zhou 2007], text categorization [Schapire and Singer 2000;

Author's address: Y. Zhang and Z.-H. Zhou, National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China. email: {zhangyin, zhouzh}@lamda.nju.edu.cn.
This research was supported by the National Science Foundation of China (60635030, 60721002), the Jiangsu Science Foundation (BK2008018), the National Fundamental Research Program of China (2010CB327903) and the Jiangsu 333 High-Level Talent Cultivation Program.

Yu et al. 2005; Zhang and Zhou 2006], scene classification [Boutell et al. 2004; Zhou and Zhang 2007], image and video annotation [Kang et al. 2006; Qi et al. 2007], bioinformatics [Elisseeff and Weston 2002; Barutcuoglu et al. 2006], association rule mining [Thabtah et al. 2004], *etc.*

The *curse of dimensionality* often causes serious problems when learning with high-dimensional data, and thus a lot of dimensionality reduction methods have been developed. Depending on whether the label information is used, those methods can be classified into two categories, i.e., *unsupervised* and *supervised*. A representative of unsupervised dimensionality reduction methods is PCA [Jolliffe 1986], which aims at identifying a lower-dimensional space maximizing the variance among data. Some recent advances include nonlinear dimensionality reduction methods such as ISOMAP [Tenenbaum et al. 2000], LLE [Roweis and Saul 2000], Laplacian Eigenmap [Belkin and Niyogi 2002], LPP [He and Niyogi 2004], *etc.*, which aim at preserving the manifold structure. A representative of supervised dimensionality reduction methods is LDA [Fisher 1936], which aims at identifying a lower-dimensional space minimizing the inter-class similarity while maximizing the intra-class similarity simultaneously. Other popular supervised dimensionality reduction methods include PLS [Wold 1985], CCA [Hardoon et al. 2004], *etc.* PLS finds orthogonal projection directions for the input by maximizing its covariance with the output, while CCA extracts the representation of the object by correlating linear relationships between two views of the object.

In spite of the fact that multi-label learning tasks usually involve high-dimensional data, multi-label dimensionality reduction remains almost untouched. In this paper, we propose a multi-label dimensionality reduction method called MDDM (Multi-label Dimensionality reduction via Dependence Maximization) which tries to identify a lower-dimensional space maximizing the dependence between the original feature description and class labels associated with the same object. We adopt the *Hilbert-Schmidt Independence Criterion* (HSIC) [Gretton et al. 2005] to measure the dependence, considering its simplicity and neat theoretical properties such as exponential convergence. We derive an eigen-decomposition problem for MDDM, which makes the multi-label dimensionality reduction process both effective and efficient. The superior performance of the proposed MDDM method is validated by experiments on a diversity of multi-label tasks.

The rest of the paper is organized as follows. In Section 2 we reviews some related work. Then, we present the MDDM method and report on experiments in Section 3 and 4, respectively, which is followed by the conclusion in Section 5.

2. RELATED WORK

An intuitive approach to multi-label learning is to decompose the task into a number of binary classification problems, each for one class [Joachims 1998; Yang 1999; Boutell et al. 2004]. For example, Boutell *et al.* [2004] applied multi-label learning techniques to scene classification. They decomposed the multi-label learning problem into multiple independent binary classification problems (one per category). In each classification problem, examples associated with the corresponding category are regarded as positive and other examples are regarded as negative. They also provided various labeling criteria to predict a set of categories for each test instance based on its output on each binary classifier. Such methods, however, usually suffer from the deficiency that the correlation among the class labels is not taken into account. Another deficiency lies in the fact that when there are lots of class labels, the number of binary classifiers needs to be generated may be too large, which may cause the problems of sparse training samples and imbalance class distribution. Zhang and Zhou [2007] extended the lazy learning algorithm, kNN, to a multi-label version, ML-kNN. It employs label prior probabilities gained from each example's k nearest neighbors and uses maximum *a posteriori* (MAP) principle to determine labels. This method overcomes the problem of imbalance class distribution contrary to the original kNN. Many other multi-label learning methods try to exploit the correlation among the class labels. Examples include methods based on probabilistic generative models [McCallum 1999; Ueda and Saito 2003], maximum entropy methods [Ghamrawi and McCallum 2005; Zhu et al. 2005], and several other recent methods. Liu *et al.* [2006] presented a semi-supervised multi-label learning method to exploit unlabeled data as well as category correlations, based on constrained non-negative matrix factorization. Correlative Multi-Label (CML), proposed in [Qi et al. 2007], simultaneously models both the individual labels and their interactions in a single formulation. Sun *et al.* [2008] employed hypergraph spectral learning to solve multi-label problems.

Some multi-label learning methods work by transforming the task into a ranking problem, trying to rank the labels in such an order that for each object, the proper labels are ranked before the other labels. Representative methods of this category include BoosTexter [Schapire and Singer 2000], RankSVM [Elisseeff and Weston 2002], *etc.*. BoosTexter is extended from the popular ensemble learning method AdaBoost. In the training phase, BoosTexter maintains a set of weights over both training examples and their labels, where training examples and their corresponding labels that are hard (easy) to predict correctly get incrementally higher (lower) weights. RankSVM defines a specific cost function and the corresponding margin for multi-label models in order to solve multi-label problems.

Other related works include multi-task learning [Bakker and Heskes 2003; Ando and Zhang 2005], which learns many related tasks together by exploring their dependency. Recently there are some works on multi-task feature selection [Obozinski et al. 2006; Argyriou et al. 2008; Liu et al. 2009]. Typically, the $l_{2,1}$-norm regularization is used to select a set of common features shared across all the tasks.

As mentioned before, although many multi-label learning tasks involve high-dimensional data, few works address the problem of multi-label dimensionality reduction. Direct application of existing dimensionality reduction methods to multi-label tasks could not result in good performance. As for unsupervised dimensionality reduction methods, they do not take labels into account and thus the label information in multi-label tasks is ignored. As for LDA, one possible way to extend to multi-label learning is to treat every combination of labels as a class. Such an extension, however, suffers from the explosion of the possible combinations of labels (e.g., for n class labels there are 2^n number of combinations) and thus is not feasible when there are lots of labels. CCA can be applied to multi-label dimensionality reduction if the features and the labels are treated as the two views of the object. When extending CCA to kernel CCA, however, a regularization term is needed to avoid trivial solutions [Bach and Jordan 2002]. PLS, similar to CCA, also ignores the correlation between labels and it could not find a space with a larger dimensionality than the number of labels [Tikhonov and Arsenin 1977]. To the best of our knowledge, the only relevant work is the MLSI method described in [Yu et al. 2005], also known as MORP in [Yu et al. 2006]. This is a multi-label extension of Latent Semantic Indexing (LSI), a popular method in information retrieval. MLSI obtains a new feature space which captures both the information of the original feature space and the label space. It has been shown that MLSI works well on a number of tasks [Yu et al. 2005; Yu et al. 2006].

3. THE MDDM METHOD

3.1 Uncorrelated Projection Dimensionality Reduction

3.1.1 *Linear case.* Let $\mathcal{X} = \mathbb{R}^D$ denote the feature space and there is a label set Θ including M labels. The proper labels associated with an instance $\boldsymbol{x}$ constitute a subset of Θ, which can be represented as a M-dimensional binary vector $\boldsymbol{y}$, with 1 indicating that the instance has the corresponding label and 0 otherwise. All the possible outputs constitute the output space $\mathcal{Y} = \{0,1\}^M$. Given a multi-label data set $\mathcal{S} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \cdots, (\boldsymbol{x}_N, \boldsymbol{y}_N)\}$, the goal is to learn from $\mathcal{S}$ a function $h : \mathcal{X} \to \mathcal{Y}$ which is able to predict proper labels for unseen instances.

Motivated by the consideration that there should exist some relation between the feature description and the labels associated with the same object, we attempt to find a lower-dimensional feature space in which the dependence between the input and output is maximized. Denote the projection vector as $\boldsymbol{p}$. An instance $\boldsymbol{x}$ is projected into a new space by $\phi(\boldsymbol{x}) = \boldsymbol{p}^{\mathrm{T}}\boldsymbol{x}$ and the kernel function induced by this space is

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) \triangleq \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle = \langle \boldsymbol{p}^{\mathrm{T}}\boldsymbol{x}_i, \boldsymbol{p}^{\mathrm{T}}\boldsymbol{x}_j \rangle. \tag{1}$$

For the output, $\boldsymbol{y} \in \mathcal{Y}$, we first consider the simplest kernel function, i.e., the linear kernel

$$l(\boldsymbol{y}_i, \boldsymbol{y}_j) \triangleq \langle \boldsymbol{y}_i, \boldsymbol{y}_j \rangle. \tag{2}$$

Those more comprehensive kernels for $\mathcal{Y}$ and their effect in the dimensionality reduction will be discussed in Section 3.3. Given $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), \cdots, (\boldsymbol{x}_N, \boldsymbol{y}_N)\}$ with joint distribution $\mathbf{P}_{\boldsymbol{xy}}$, we can define the kernel matrix for feature space as $\mathbf{K} = [K_{ij}]_{N\times N}$, $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and the kernel matrix for label space as $\mathbf{L} = [L_{ij}]_{N\times N}$, $L_{ij} = l(\boldsymbol{y}_i, \boldsymbol{y}_j)$. Then, we try to maximize the dependence between the feature description and the class labels.

The Hilbert-Schmidt Independence Criterion, HSIC [Gretton et al. 2005], computes the square of the norm of the cross-covariance operator over the domain $\mathcal{X} \times \mathcal{Y}$ in Hilbert Space, which has been applied to several machine learning tasks recently as a useful measure of dependence [Song et al. 2007; Song et al. 2008]. Due to its simplicity and neat theoretical properties, we employ it here for our purpose. An empirical estimate of HSIC [Gretton et al. 2005] is

$$\mathrm{HSIC}(\mathcal{F}, \mathcal{Y}, \mathbf{P}_{\boldsymbol{xy}}) = (N-1)^{-2}\mathrm{tr}\left(\mathbf{HKHL}\right), \tag{3}$$

where $\mathrm{tr}(\cdot)$ is the trace of a matrix and $\mathbf{H} = I - \frac{1}{N}\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}$, $\boldsymbol{e}$ is a all-one column vector.

Since the normalization term in Eq. 3 does not affect the optimization procedure, we can drop it and only consider $\mathrm{tr}\,(\mathbf{HKHL})$. Denoting $X = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N]$ and $Y = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_N]$, we have $\phi(X) = \boldsymbol{p}^{\mathrm{T}} X$, $\mathbf{K} = \langle \phi(X), \phi(X) \rangle = X^{\mathrm{T}} \boldsymbol{p}\boldsymbol{p}^{\mathrm{T}} X$ and $\mathbf{L} = Y^{\mathrm{T}} Y$. We can rewrite the optimization procedure as searching for the optimal linear projection

$$\boldsymbol{p}^* = \arg\max_{\boldsymbol{p}} \ \mathrm{tr}\left(\mathbf{H} X^{\mathrm{T}} \boldsymbol{p}\boldsymbol{p}^{\mathrm{T}} X \mathbf{HL}\right) . \tag{4}$$

To avoid the scaling problem, we add the constraint that the l_2-norm of $\boldsymbol{p}$ should be 1. Therefore, we get the optimization problem

$$\begin{cases} \max\limits_{\boldsymbol{p}} & \mathrm{tr}\left(\mathbf{H} X^{\mathrm{T}} \boldsymbol{p}\boldsymbol{p}^{\mathrm{T}} X \mathbf{HL}\right) \\ \text{s.t.} & \boldsymbol{p}^{\mathrm{T}}\boldsymbol{p} = 1 \ . \end{cases} \tag{5}$$

Notice that

$$\mathrm{tr}\left(\mathbf{H} X^{\mathrm{T}} \boldsymbol{p}\boldsymbol{p}^{\mathrm{T}} X \mathbf{HL}\right) = \boldsymbol{p}^{\mathrm{T}} \left(X\mathbf{HLH}X^{\mathrm{T}}\right) \boldsymbol{p} \ . \tag{6}$$

Since $X\mathbf{HLH}X^{\mathrm{T}}$ is symmetric, the eigenvalues are all real. If the eigenvalues of $X\mathbf{HLH}X^{\mathrm{T}}$ are sorted as $\lambda_1 \geq \cdots \geq \lambda_D$, then the optimal $\boldsymbol{p}^*$ is the normalized eigenvector corresponding to the largest eigenvalue, λ_1.

To find the following projection direction which maximizes the correlation between the feature space and label space, we require that it should be orthonormal to the previous projection directions. Suppose we want to reduce the original space to a d-dimensional space, $\mathcal{F}$, and denote $P = [\boldsymbol{p}_1, \cdots, \boldsymbol{p}_d]$ $(d \ll D)$. As the previous requirement, $\boldsymbol{p}_i^{\mathrm{T}} \boldsymbol{p}_j = \delta_{ij}$ $(1 \leq i, j \leq d)$, where

$$\delta_{ij} = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{if} \quad i \neq j \ . \end{cases} \tag{7}$$

Therefore, $\boldsymbol{p}_i^*$'s $(1 \leq i \leq d)$ are the normalized eigenvectors corresponding to the largest d eigenvalues, i.e., from λ_1 to λ_d, and they form the basis spanning the new space $\mathcal{F}$. Note that if the rank of $\mathbf{L}$ is r then λ_i $(i > r)$ are all zeros. If the projection P^* has been obtained, then the corresponding HSIC value is

$$\mathrm{HSIC} = \sum_{i=1}^{d} \lambda_i \ . \tag{8}$$

Since the eigenvalues reflect the contribution of the corresponding dimensions, we can control d by setting a threshold thr $(0 \leq thr \leq 1)$ and then choose the first d eigenvectors such that

$$\sum_{i=1}^{d} \lambda_i \geq thr \times \left(\sum_{i=1}^{D} \lambda_i \right) \ . \tag{9}$$

Thus, the optimization problem reduces to deriving eigenvalues of a $D \times D$ matrix and the computational complexity is $O(dD^2)$ for obtaining the largest d eigenvalues. The Pseudo-code of the MDDM method is shown in Fig. 1.

Note that HSIC is just one among the many choices we can take to measure the dependence. Other measures, such as Kullback-Leibler divergence [Davis et al. 2007] can also be applied to the MDDM method with careful design. In this paper we focus on HSIC and the study on alternative measures is left for future work.

```
MDDM($X$, $Y$, $d$ or $thr$)
Input:
  $X$  : $D \times N$ feature matrix
  $Y$  : $M \times N$ label matrix
  $d$  : the dimensionality to be reduced to
  $thr$: a threshold
Process:
  1  Construct the label kernel matrix $\mathbf{L}$.
  2  Calculate $X\mathbf{HLH}X^{\mathrm{T}}$.
  3  if $d$ is given
  4     Do eigen-decomposition on $X\mathbf{HLH}X^{\mathrm{T}}$, then construct $D \times d$ matrix
        $P^*$ whose columns are composed by the eigenvectors corresponding to
        the largest $d$ eigenvalues.
  5  else (i.e., $thr$ is given)
  6     Construct $D \times r$ matrix $\tilde{P}^*$ in a way similar to Step 4 where $r$ is the
        rank of $\mathbf{L}$, then choose the first $d$ eigenvectors that enable $\sum_{i=1}^{d} \lambda_i \geq$
        $thr \times \left( \sum_{i=1}^{D} \lambda_i \right)$ to compose $P^*$.
  7  end if
Output:
  $P^*$: the projection from $\mathbb{R}^D$ to $\mathbb{R}^d$
```

Fig. 1. Pseudo-code of the MDDM method

3.1.2 *Nonlinear case.* In this subsection, we consider the problem of nonlinear multi-label dimensionality reduction. Firstly an instance $\boldsymbol{x}$ is mapped into a *Reproducing Kernel Hilbert Space* (RKHS) $\mathcal{Q}$ by a nonlinear positive semi-definite function $\varphi : \boldsymbol{x} \mapsto \varphi(\boldsymbol{x})$ and $\mathbf{Q}$ is the corresponding kernel matrix of $\mathcal{Q}$ for training data, i.e., $\mathbf{Q}_{ij} = \langle \varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j) \rangle$. Denote the subspace spanned by $\varphi(\boldsymbol{x}_i)$, $1 \leq i \leq N$, as Φ. Then for all $\boldsymbol{q} \in \mathcal{Q}$, it can be expressed as $\boldsymbol{q} = \sum_{i=1}^{N} c_i \varphi(\boldsymbol{x}_i) + \boldsymbol{q}^{\perp}$ where $\boldsymbol{q}^{\perp}$ is orthogonal to Φ. The projection of $\varphi(\boldsymbol{x}_i)$ by $\boldsymbol{q}$ can be written as

$$\phi(\boldsymbol{x}_i) = \langle \varphi(\boldsymbol{x}_i), \boldsymbol{q} \rangle = \left\langle \varphi(\boldsymbol{x}_i), \sum_{j=1}^{N} c_j \varphi(\boldsymbol{x}_j) \right\rangle = \sum_{j=1}^{N} c_j \left\langle \varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j) \right\rangle = \boldsymbol{c}^{\mathrm{T}} \mathbf{Q}_{\cdot i} \quad (10)$$

where $\boldsymbol{c} = [c_1, c_2, \cdots, c_N]^{\mathrm{T}}$ and $\mathbf{Q}_{\cdot i}$ is the i-th column of $\mathbf{Q}$. Similar to kernel PCA, we only need to consider the subspace of $\mathcal{Q}$ that contains the span of the data, i.e., $\boldsymbol{q}$ can be expressed as $\boldsymbol{q} = \sum_{i=1}^{N} c_i \varphi(\boldsymbol{x}_i)$.

Then, the kernel function between two instances in the projected feature space is

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left\langle \langle \varphi(\boldsymbol{x}_i), \boldsymbol{q} \rangle, \langle \varphi(\boldsymbol{x}_j), \boldsymbol{q} \rangle \right\rangle = \left\langle \boldsymbol{c}^{\mathrm{T}} \mathbf{Q}_{\cdot i}, \boldsymbol{c}^{\mathrm{T}} \mathbf{Q}_{\cdot j} \right\rangle . \quad (11)$$

Therefore, the kernel matrix of feature space can be expressed as $\mathbf{K} = \mathbf{Q}\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}}\mathbf{Q}$. The l_2-norm constraint on $\boldsymbol{q}$ becomes $\|\boldsymbol{q}\|_2 = \boldsymbol{c}^{\mathrm{T}}\mathbf{Q}\boldsymbol{c} = 1$. Thus we have the optimization problem

$$\begin{cases} \max\limits_{\boldsymbol{c}} & \mathrm{tr}\left(\mathbf{HQ}\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}}\mathbf{QHL}\right) \\ \text{s.t.} & \boldsymbol{c}^{\mathrm{T}}\mathbf{Q}\boldsymbol{c} = 1 \ . \end{cases} \quad (12)$$

Since $\mathrm{tr}\left(\mathbf{HQ}\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}}\mathbf{QHL}\right) = \boldsymbol{c}^{\mathrm{T}}\mathbf{QHLHQ}\boldsymbol{c}$, by Lagrange method, this optimization problem is equivalent to the general eigen-decomposition problem

$$\mathbf{QHLHQ}\boldsymbol{c} = \lambda\mathbf{Q}\boldsymbol{c} \ . \tag{13}$$

Thus, the optimal $\boldsymbol{c}^*$ is the eigenvector corresponding to the largest eigenvalue of the general eigen-decomposition problem.

For an unseen instance $\boldsymbol{x}'$, the projection $\phi(\boldsymbol{x}')$ is expressed as

$$\phi(\boldsymbol{x}') = \langle\varphi(\boldsymbol{x}'), \boldsymbol{q}\rangle = \sum_{i=1}^{N} c_i \left\langle\varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}')\right\rangle = \boldsymbol{c}^{\mathrm{T}}k(X, \boldsymbol{x}') \tag{14}$$

where $k(X, \boldsymbol{x}')$ is $[k(\boldsymbol{x}_1, \boldsymbol{x}'), \cdots, k(\boldsymbol{x}_N, \boldsymbol{x}')]^{\mathrm{T}}$.

If the reduced feature space, $\mathcal{F}$, has dimension d, we can get the coefficient matrix $C = [\boldsymbol{c}_1, \cdots, \boldsymbol{c}_d]$ with the constraint $\boldsymbol{c}_i^{\mathrm{T}}\mathbf{Q}\boldsymbol{c}_j = \delta_{ij}$. Similar to the linear case we know $\boldsymbol{c}_i$ is the eigenvector corresponding to the i-th largest eigenvalue of the general eigen-decomposition problem. The HSIC between feature space and label space is the sum of the largest d eigenvalues. Since $\mathbf{QHLHQ}$ is an $N \times N$ matrix, the computational complexity is $O(dN^2)$ for obtaining the largest d eigenvalues. If $N < D$, linear dimensionality reduction can also adopt this formulation where $\mathbf{Q} = X^{\mathrm{T}}X$.

3.2 Uncorrelated Subspace Dimensionality Reduction

3.2.1 *Linear case.* In Section 3.1, we study MDDM to find a subspace when the projection directions are orthonormal, i.e., $\boldsymbol{p}_i^{\mathrm{T}}\boldsymbol{p}_j = \delta_{ij}$. Such projection strategy, however, will still remain some redundant information in the lower-dimensional data set [Chen et al. 2008]. In order to remove that redundant information in the reduced subspace, we hope that instances in the lower-dimensional space have uncorrelated features.

Let's still use P as the projection matrix from the original space to a lower-dimensional space $\mathcal{F}$. Here we assume that we have centered data, i.e. $\sum_i \boldsymbol{x}_i = 0$. It can be achieved by a simple axis translation. We want the projected features to be uncorrelated, i.e. $\mathrm{Cor}(\boldsymbol{p}_i^{\mathrm{T}}X, \boldsymbol{p}_j^{\mathrm{T}}X) = \delta_{ij}$ for $1 \le i, j \le d$. Thus, we get the new constraint,

$$\boldsymbol{p}_i^{\mathrm{T}}XX^{\mathrm{T}}\boldsymbol{p}_j = \delta_{ij} \quad (1 \le i, j \le d) \ . \tag{15}$$

The projected features found uncorrelated on the training set are probably not uncorrelated on the test set. For generality, we add the regularization term to the constraint, i.e., $\boldsymbol{p}_i^{\mathrm{T}}\left(\mu XX^{\mathrm{T}} + (1-\mu)I\right)\boldsymbol{p}_j = \delta_{ij}$, where $\mu \in [0, 1]$ is a pre-defined parameter to control the importance between two constraints. Note when $\mu = 0$, the constraint requires the projections to be orthonormal and when $\mu = 1$, the projected features are uncorrelated on the training data. Based on the basic assumption that the required projection P should make the input space have high dependence with the output space, the optimization problem of uncorrelated subspace dimensionality reduction can be written as

$$\begin{cases} \max\limits_{P} & \mathrm{tr}\left(\mathbf{H}X^{\mathrm{T}}PP^{\mathrm{T}}X\mathbf{HL}\right) \\ \text{s.t.} & \boldsymbol{p}_i^{\mathrm{T}}\left(\mu XX^{\mathrm{T}} + (1-\mu)I\right)\boldsymbol{p}_j = \delta_{ij} \quad (1 \le i, j \le d) \ . \end{cases} \tag{16}$$

From the analysis in Section 3.1, $\boldsymbol{p}_i^*$ is the eigenvector corresponding to the i-th largest eigenvalue of the general eigen-decomposition problem

$$X\mathbf{HLH}X^{\mathrm{T}}\boldsymbol{p} = \lambda\left(\mu XX^{\mathrm{T}} + (1-\mu)I\right)\boldsymbol{p} \ . \tag{17}$$

For clarity, we denote the MDDM method with uncorrelated projection constraint, $P^{\mathrm{T}}P = I$, as MDDM$_p$, and that with uncorrelated feature constraint, $P\left(\mu XX^{\mathrm{T}} + (1-\mu)I\right)P^{\mathrm{T}} = I$, as MDDM$_f$.

3.2.2 *Nonlinear case.* Similar to Subsection 3.1.2, we can extend linear MDDM$_f$ to the nonlinear case. Also assume $\boldsymbol{x}$ is first imaged into an RKHS $\mathcal{Q}$ by φ and then projected via a function $\boldsymbol{q} \in \mathcal{Q}$ as $\phi(\boldsymbol{x}) = \langle\varphi(\boldsymbol{x}), \boldsymbol{q}\rangle$. Similar to the linear case, here we assume $\mathbf{Q}$ is centered, $\mathbf{Q}\boldsymbol{e} = \boldsymbol{e}^{\mathrm{T}}\mathbf{Q} = 0$. As the discussion in Section 3.1.2, $\boldsymbol{q} = \sum_{i=1}^{N} c_i\varphi(\boldsymbol{x}_i)$. Therefore, the uncorrelated requirement in Eq. 15 is rewritten as $\boldsymbol{c}_i^{\mathrm{T}}\mathbf{QQ}\boldsymbol{c}_j = \delta_{ij}$. Similarly, a regularization term is added for generality and the constraint becomes

$$\boldsymbol{c}_i^{\mathrm{T}}\left(\mu\mathbf{QQ} + (1-\mu)\mathbf{Q}\right)\boldsymbol{c}_j = \delta_{ij} \ . \tag{18}$$

The optimization problem now becomes

$$\begin{cases} \max\limits_{C} & \mathrm{tr}\left(\mathbf{HQ}^{\mathrm{T}}CC^{\mathrm{T}}\mathbf{QHL}\right) \\ \text{s.t.} & \boldsymbol{c}_i^{\mathrm{T}}\left(\mu\mathbf{QQ} + (1-\mu)\mathbf{Q}\right)\boldsymbol{c}_j = \delta_{ij} \quad (1 \le i, j \le d) \ . \end{cases} \tag{19}$$

Finally, we can get $\boldsymbol{c}_i^*$ which is the eigenvector corresponding to the i-th largest eigenvalue of the general eigen-decomposition problem

$$\mathbf{QHLHQ}\ \boldsymbol{c} = \lambda\left(\mu\mathbf{QQ} + (1-\mu)\mathbf{Q}\right)\boldsymbol{c} \ . \tag{20}$$

3.3 Taking Label Correlation into Consideration

Note that in the above analysis, we used the inner product in the original label space $\mathcal{Y}$ as the label kernel function, i.e., $l(\boldsymbol{y}_i, \boldsymbol{y}_j) = \langle\boldsymbol{y}_i, \boldsymbol{y}_j\rangle$. This simple linear kernel does not take the correlation between labels into consideration. For that purpose, we can define more complex kernel function. If $\boldsymbol{y}$ is projected by an intrinsic mapping function π to a new RKHS $\mathcal{G}$ with the kernel function $l(\boldsymbol{y}_i, \boldsymbol{y}_j) = \langle\pi(\boldsymbol{y}_i), \pi(\boldsymbol{y}_j)\rangle$, then the HSIC of $\boldsymbol{x}$ and $\boldsymbol{y}$ with kernel function $k(\cdot,\cdot)$ and $l(\cdot,\cdot)$ is $\mathrm{HSIC}(\mathcal{F}, \mathcal{G}, \mathbf{P}_{\boldsymbol{xy}}) = \mathrm{tr}\left(\mathbf{HKHL}\right)$. For example, to take the correlation of labels into consideration, $\mathbf{B} = [B_{st}]_{M\times M}$ can be introduced, where B_{st} $(1 \le s, t \le M)$ indicates the correlation between labels θ_s and θ_t. Then, the label kernel matrix is obtained by $\mathbf{L} = Y^{\mathrm{T}}\mathbf{B}Y$. In our previous discussion the labels are assumed to be independent, thus $\mathbf{B} = I$ and $\mathbf{L} = Y^{\mathrm{T}}Y$. If $\mathbf{B}$ can really reflect the correlation between labels, the dimensionality reduction on feature space could derive better results. However, how to measure the correlation between labels, i.e., to get a proper $\mathbf{B}$, is still a difficult problem beyond our discussion. One possible way was proposed in [Liu et al. 2006] where each label θ_s was treated as an $N \times 1$ binary vector, $\boldsymbol{\theta}_s$, with the i-th element θ_s^i indicating whether $\boldsymbol{x}_i$ has the label θ_s, and then rbf kernel was used to get $\mathbf{B}$.

4. EXPERIMENTS

In our experiments, we compare our MDDM methods, MDDM$_p$ and MDDM$_f$, with other dimensionality reduction methods, including PCA, LPP, PLS, CCA and

MLSI. In MDDM$_f$, μ is set as 0.5. In LPP, the number of nearest neighbors used for constructing *adjacency graph* is as same as that used in ML-kNN for classification. In kernel CCA, the regularization parameter is set to be 0.5. In MLSI, the parameter controlling the tradeoff between feature and label is set to 0.5 as recommended in [Yu et al. 2005; Yu et al. 2006]. The dimensionality of the lower-dimensional space, d, is decided by setting $thr = 99.9\%$ in Fig. 1. All dimensionality reduction methods reduce to the same dimensionality. The performance under different d values will be reported later in this section. We also compare with a simple multi-label feature selection method, denoted as SEL, which sequentially selects the most discriminative features. Here the discriminative capability is defined as same as that used in LDA. For the k-th feature of $\boldsymbol{x}$, x^k, its discriminative capability can be expressed as

$$\frac{\sum_{m=1}^{M}\sum_{\boldsymbol{x}\in\Omega_m}\left(x^k-\Delta_m^k\right)^2}{\sum_{m=1}^{M}\left(\Delta_m^k-\Delta^k\right)^2} \tag{21}$$

where Ω_m is the subset containing all the instances with the m-th label, N_m is the cardinality of Ω_m, $\Delta_m^k = \big(\sum_{\boldsymbol{x}\in\Omega_m} x^k\big)/N_m$, $\Delta^k = \big(\sum_{m=1}^{M}\sum_{\boldsymbol{x}\in\Omega_m} x^k\big)/\big(\sum_{m=1}^{M} N_m\big)$. The number of selected features is d. We use the performance in the original space as the baseline and denoted as ORI. The base classifiers used are ML-kNN with default setting $k = 10$ [Zhang and Zhou 2007] and binary SVM with the regularization parameter C selected from the set $\{10^t\}_{t=-2}^{2}$ by 5-fold cross validation on training set.

4.1 Application Tasks

We consider three real application tasks, including web page classification, image annotation and text categorization.

4.1.1 *Web page classification.* First, a collection of eleven data sets[1] are used in this experiment. Using the "Bag-of-Words" representation [Dumais et al. 1998], Ueda and Saito [2003] tried to categorize real Web pages linked from the "yahoo.com" domain, which consists of 14 top-level categories (i.e. "Arts&Humanities", "Business&Economy", *etc.*) and each category is classified into a number of second-level subcategories. By focusing on the second-level categories, they tested 11 out of the 14 independent text categorization problems. For each problem, the training set contains 2,000 documents while the test set contains 3,000 documents. About $20\% \sim 45\%$ of them belong to multiple subcategories simultaneously. In this task, the kernel for MDDM$_p$, MDDM$_f$, PCA, CCA, PLS and binary SVM is linear kernel.

4.1.2 *Image annotation.* The second task is automatic image annotation on Corel database[2]. Each image has been segmented into several regions and tagged with several words. The regions with similar features are clustered into 500 clusters, known as blobs [Duygulu et al. 2002]. Each image is then represented by a 500-dimensional binary vector with each dimension indicating whether the corresponding cluster appears in the image. 374 words are used for annotation and the

[1] `http://www.kecl.ntt.co.jp/as/members/ueda/yahoo.tar.gz`

[2] `http://kobus.ca/research/data/eccv_2002`

words with less than 50 positive assignments are filtered out. Each remaining annotation word is regarded as a label. Details of the data set can be found in [Duygulu et al. 2002; Kang et al. 2006]. The original data set is divided into a training set containing 4,500 images and a test set containing 500 images. In this experiment, we merge them into one data set containing 5,000 images and then perform 5-fold cross-validation. In this task, the kernel for MDDM$_p$, MDDM$_f$, PCA, CCA, PLS and binary SVM is rbf kernel. The kernel width is selected by 5-fold cross validation from the set $\{10^{0.5t}\}_{t=-2}^{2}$ on training set.

4.1.3 *Text categorization.* In this experiment, we perform text categorization using rcv1v2 database [Lewis et al. 2004]. Here we use the five subsets of the rcv1v2 data set[3]. Each subset contains a training set and a test set, each including 3,000 documents. On every subset, features with less than five occurrences and topics with less than 50 positive assignments are filtered out. Each remaining topic is treated as a label. Around 4,000 features and 50 labels are left. Note that the number of examples in this subset (6,000) is much larger than in previous tasks in this paper, and the dimensionality (4,238) is also very high. In this task, the kernel for MDDM$_p$, MDDM$_f$, PCA, CCA, PLS and binary SVM is linear kernel.

4.2 Evaluation Metrics

Multi-label learning systems require more complicated evaluation criteria than traditional single-label systems. In this section we briefly summarize the criteria used for performance evaluation from various perspectives. In this paper we employ two sets of criteria to evaluate the performance of *label set prediction* as well as the performance of *label ranking*.

The first group of evaluation criteria concern with the performance on label set prediction for each instance, based on the label prediction function $h : \mathcal{X} \to \mathcal{Y}$ of each algorithm. Let $h(\boldsymbol{x}_i)$ be the binary label vector predicted by an multi-label classifier for instance $\boldsymbol{x}_i$.

(1) *Hamming loss (HL)*: Evaluates how many times an instance-label pair is misclassified.

$$HL(h, S) = \frac{1}{N} \sum_{i=1}^{N} \frac{\|h(\boldsymbol{x}_i) \oplus \boldsymbol{y}_i\|_1}{M}$$

where $\oplus$ stands for the XOR operation and $\|\cdot\|_1$ is the l_1-norm. The smaller the value, the better the performance. This is one of the most important multi-label criteria, and has been used in many works.

(2) *Macro-F1 (macroF1)*: Averages the *F1* measure on the predictions of different labels.

$$macroF1(h, S) = \frac{1}{M} \sum_{m=1}^{M} \frac{2 \times \sum_{i=1}^{N} h^m(\boldsymbol{x}_i)\boldsymbol{y}_i^m}{\sum_{i=1}^{N} \boldsymbol{y}_i^m + \sum_{i=1}^{N} h(\boldsymbol{x}_i)^m}$$

where $\boldsymbol{y}^m$ is the m-th element of $\boldsymbol{y}$ and $h^m(\boldsymbol{x})$ is the m-th element of $h^m(\boldsymbol{x})$. The larger the value, the better the performance.

[3] `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multi-label.html`

(3) *Micro-F1 (microF1)*: Calculates the *F1* measure on the predictions of different labels as a whole.

$$microF1(h,S) = \frac{2 \times \sum_{i=1}^{N} \|h(\boldsymbol{x}_i) \bigcap \boldsymbol{y}_i\|_1}{\sum_{i=1}^{N} \|\boldsymbol{y}_i\|_1 + \sum_{i=1}^{N} \|h(\boldsymbol{x}_i)\|_1}$$

The larger the value, the better the performance.

The second group of evaluation criteria concern with the label ranking performance for each instance, based on the real-valued output function $f : \mathcal{X} \times \Theta \rightarrow \mathbb{R}$ of each algorithm. $f(\cdot,\cdot)$ can be transformed into a ranking function $rank_f(\cdot,\cdot)$, which maps the outputs of $f(\boldsymbol{x}_i, \theta_s)$ for any $\theta_s \in \Theta$ to $\{1, 2, \cdots, M\}$ such that if $f(\boldsymbol{x}_i, \theta_s) > f(\boldsymbol{x}_i, \theta_t)$ then $rank_f(\boldsymbol{x}_i, \theta_s) < rank_f(\boldsymbol{x}_i, \theta_t)$. We use Θ_i to indicate the subset of Θ corresponding to $\boldsymbol{y}_i$. These criteria have been used in [Schapire and Singer 2000; Elisseeff and Weston 2002; Zhang and Zhou 2007].

(4) *Ranking loss (RL)*: Evaluates the average fraction of label pairs that are not correctly ordered.

$$RL(f,S) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\Theta_i||\overline{\Theta}_i|} \Big| \{(\theta_s, \theta_t) \in \Theta_i \times \overline{\Theta}_i \big| f(\boldsymbol{x}_i, \theta_s) \leq f(\boldsymbol{x}_i, \theta_t)\} \Big|$$

where the $\overline{\Theta}_i$ denotes the complementary set of Θ_i in Θ and $|\cdot|$ is the cardinality of a set. The smaller the value, the better the performance.

(5) *Average Precision (AP)*: Evaluates the average fraction of labels ranked above a particular label $\theta \in \Theta_i$ which is actually in Θ_i.

$$AP(f,S) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\Theta_i|} \sum_{\theta \in \Theta_i} \frac{|\{\theta' \in \Theta_i | rank_f(\boldsymbol{x}_i, \theta') \leq rank_f(\boldsymbol{x}_i, \theta)\}|}{rank_f(\boldsymbol{x}_i, \theta)}$$

The larger the value, the better the performance.

(6) *One-error (OE)*: Evaluates how many times the top-ranked label is not in the set of ground-truth labels of the instance.

$$OE(f,S) = \frac{1}{N} \sum_{i=1}^{N} \delta \left(\left[\arg\max_{\theta \in \Theta} f(\boldsymbol{x}_i, \theta) \right] \notin \Theta_i \right)$$

where $\delta(z)$ is the indicator function which equals 1 if z holds and 0 otherwise. The smaller the value, the better the performance.

(7) *Coverage (CV)*: Evaluates how far, on average, we need to go down the label ranking list to cover all the ground-truth labels of the instance.

$$CV(f,S) = \frac{1}{N} \sum_{i=1}^{N} \max_{\theta \in \Theta_i} rank_f(\boldsymbol{x}_i, \theta) - 1$$

The smaller the value, the better the performance.

To evaluate the ranking performance under each label, we also study the *Area Under ROC Curves* (AUC) for each label. We treat the multi-label problem as M binary classification problems and calculate AUC for each problem. Then the average AUC over all the labels is recorded.

Note that the eight criteria evaluate the performance of multi-label learning systems from different aspects. Usually few algorithms could outperform other algorithms on all those criteria.

4.3 Experimental Results

First, we set the label kernel as linear and the influence of the label kernel will be studied in Section 4.4.

The results of the compared methods on the eleven data sets in the task of web page classification are summarized in Table I and II where ML-kNN and SVM are used as the base classifier, respectively. On each evaluation criterion, the best result and the results comparable with the best one are highlighted in boldface. Here the comparability is checked by pairwise t-tests at 95% significance level. From the tables we can see that when the base classifier is ML-kNN, MDDM$_p$ is significantly better than all previous methods on all criteria and MDDM$_f$ also performs quite well on all criteria except Hamming Loss. When the base classifier is SVM, MDDM$_p$ is significantly better than all previous methods on all criteria except Micro-F1 and MDDM$_f$ performs quite well on Micro-F1 and Macro-F1.

The results of the compared methods on the task of image annotation are tabulated in Table III and IV where ML-kNN and SVM are used as the base classifier, respectively. On this task, with either base classier, MDDM methods are among the best performance methods.

The results of the compared methods on the task of text categorization are shown in Table V and VI where ML-kNN and SVM are used as the base classifier, respectively. On this task, MDDM$_p$ is significantly better than most previous methods on all the evaluation criteria, no matter which base classifier is used. MDDM$_f$, though slightly inferior to MDDM$_p$, performs also quite well on most criteria.

4.4 Embedding the Label Relationship

In this section, we take the label relationship into consideration. According to the discussion in Section 3.3, we use the matrix $\mathbf{B}$ to reflect the relationship between labels and the label kernel matrix is computed as $\mathbf{L} = Y^{\mathrm{T}}\mathbf{B}Y$. Here, similar to [Liu et al. 2006], we assume that $\mathbf{B}$ is induced from rbf kernel and the kernel width is selected via 5-fold cross-validation from the set $\{10^{0.5t}\}_{t=-2}^{2}$. Since only our MDDM methods take label kernel into consideration, we compare MDDM$_p$ and MDDM$_f$ with and without embedding the label relationships, denoted as MDDM$_p^l$, MDDM$_f^l$ ($\mathbf{B} = I$) and MDDM$_p^k$, MDDM$_f^k$ ($\mathbf{B}$ is a rbf kernel matrix), respectively. The results on different tasks are tabulated in Table VII to XII, respectively.

From the results we can see that embedding the label relationship can improve the performance significantly in many cases, and at least will not make the performance worse. Note that the strategy we used for embedding label relationship at here is quite simple, and it is expectable that greater improvement can be obtained with better strategies. Further study on this issue is left for future work.

4.5 The Influence of d on Performance

Now we study the performance of the compared methods under different d values, i.e., the dimensionality of the lower-dimensional space.

On web page classification and image annotation, we run experiments with d

Table I. Results (mean±std.) on web page classification when ML-kNN is used as base classifier (↓ indicates "the smaller the better"; ↑ indicates "the larger the better". The best result and the results comparable with the best one are highlighted in boldface.)

Methods	HL($\times 10^1$) ↓	microF1↑	macroF1 ↑	RL ↓
MDDM$_p$	**0.419±0.128**	0.465±0.136	0.297±0.069	**0.095±0.036**
MDDM$_f$	0.453±0.137	**0.496±0.113**	**0.325±0.056**	0.099±0.037
MLSI	0.603±0.160	0.396±0.072	0.286±0.058	0.228±0.061
CCA	0.486±0.140	0.460±0.114	0.300±0.054	0.105±0.038
PLS	0.438±0.138	0.394±0.173	0.192±0.071	0.113±0.042
PCA	0.450±0.136	0.343±0.176	0.186±0.070	0.109±0.041
LPP	0.469±0.138	0.295±0.170	0.170±0.058	0.118±0.044
SEL	0.461±0.134	0.332±0.173	0.193±0.056	0.111±0.039
ORI	0.470±0.139	0.308±0.182	0.186±0.062	0.112±0.044

Methods	OE ↓	CV($\times 10^{-1}$) ↓	AP ↑	AUC ↑
MDDM$_p$	**0.407±0.134**	**0.376± 0.118**	**0.672±0.102**	**0.651±0.022**
MDDM$_f$	**0.413±0.129**	0.386± 0.121	0.665±0.100	**0.649±0.025**
MLSI	0.544±0.067	0.779± 0.192	0.510±0.065	0.633±0.019
CCA	0.456±0.138	0.402± 0.126	0.636±0.106	0.636±0.027
PLS	0.451±0.154	0.431± 0.138	0.631±0.117	0.613±0.023
PCA	0.471±0.155	0.415± 0.129	0.623±0.116	0.622±0.022
LPP	0.503±0.162	0.443± 0.132	0.597±0.120	0.584±0.030
SEL	0.484±0.153	0.422± 0.126	0.613±0.113	0.602±0.027
ORI	0.489±0.162	0.424± 0.130	0.610±0.121	0.603±0.038

Table II. Results (mean±std.) on web page classification when SVM is used as base classifier

Methods	HL($\times 10^1$) ↓	microF1 ↑	macroF1 ↑	RL ↓
MDDM$_p$	**0.411±0.126**	0.468± 0.132	0.299± 0.065	**0.147± 0.045**
MDDM$_f$	0.483±0.152	**0.493± 0.107**	**0.333± 0.053**	0.185± 0.048
MLSI	0.514±0.113	0.334± 0.077	0.281± 0.052	0.416± 0.110
CCA	0.586±0.145	0.446± 0.105	0.292± 0.060	0.204± 0.053
PLS	0.440±0.138	0.351± 0.210	0.162± 0.062	0.261± 0.076
PCA	0.454±0.138	0.295± 0.205	0.149± 0.059	0.200± 0.066
LPP	0.480±0.144	0.239± 0.214	0.141± 0.064	0.269± 0.078
SEL	0.463±0.133	0.276± 0.181	0.167± 0.061	0.231± 0.083
ORI	0.416±0.125	0.481± 0.123	0.296± 0.066	0.177± 0.049

Methods	OE ↓	CV($\times 10^{-2}$) ↓	AP ↑	AUC ↑
MDDM$_p$	**0.399± 0.128**	**0.055± 0.014**	**0.639± 0.101**	**0.716± 0.036**
MDDM$_f$	0.419± 0.123	0.069± 0.020	0.619± 0.098	0.695± 0.033
MLSI	0.600± 0.087	0.135± 0.037	0.428± 0.091	0.683± 0.026
CCA	0.457± 0.126	0.076± 0.024	0.591± 0.099	0.679± 0.033
PLS	0.467± 0.160	0.094± 0.030	0.564± 0.128	0.624± 0.020
PCA	0.512± 0.179	0.072± 0.022	0.545± 0.133	0.676± 0.045
LPP	0.539± 0.181	0.094± 0.022	0.507± 0.138	0.616± 0.041
SEL	0.544± 0.193	0.082± 0.024	0.508± 0.144	0.631± 0.037
ORI	0.411± 0.124	0.067± 0.019	**0.641± 0.098**	0.696± 0.033

Table III. Results (mean±std.) on image annotation when ML-kNN is used as base classifier (↓ indicates "the smaller the better"; ↑ indicates "the larger the better". The best result and the results comparable with the best one are highlighted in boldface.)

Methods	HL($\times 10^1$) ↓	microF1 ↑	macroF1 ↑	RL ↓
MDDM$_p$	**0.113±0.001**	0.170±0.016	**0.382±0.006**	**0.138±0.003**
MDDM$_f$	**0.113±0.000**	**0.202±0.010**	**0.383±0.002**	0.143±0.003
MLSI	0.115±0.000	0.002±0.001	0.344±0.005	0.170±0.004
CCA	0.114±0.000	0.020±0.007	0.350±0.005	0.162±0.004
PLS	0.115±0.000	0.046±0.010	0.353±0.005	0.152±0.003
PCA	0.114±0.001	0.066±0.006	0.360±0.006	0.143±0.004
LPP	0.114±0.001	0.032±0.005	0.354±0.006	0.159±0.003
SEL	0.115±0.001	0.027±0.005	0.353±0.006	0.158±0.003
ORI	0.115±0.001	0.030±0.008	0.353±0.006	0.160±0.005

Methods	OE ↓	CV($\times 10^{-3}$) ↓	AP ↑	AUC ↑
MDDM$_p$	**0.631±0.014**	**0.097±0.002**	**0.329±0.010**	**0.600±0.004**
MDDM$_f$	0.667±0.014	0.100±0.003	0.301±0.011	0.586±0.005
MLSI	0.780±0.015	0.115±0.003	0.209±0.007	0.510±0.004
CCA	0.736±0.013	0.111±0.003	0.239±0.006	0.529±0.001
PLS	0.678±0.021	0.107±0.002	0.285±0.011	0.573±0.004
PCA	0.679±0.013	0.101±0.003	0.295±0.010	0.582±0.006
LPP	0.734±0.008	0.110±0.003	0.247±0.005	0.534±0.007
SEL	0.744±0.014	0.109±0.003	0.243±0.007	0.548±0.002
ORI	0.758±0.034	0.109±0.003	0.235±0.010	0.542±0.004

Table IV. Results (mean±std.) on image annotation when SVM is used as base classifier

Methods	HL($\times 10^1$) ↓	microF1↑	macroF1 ↑	RL ↓
MDDM$_p$	0.114±0.000	**0.162±0.015**	**0.378±0.004**	**0.172±0.003**
MDDM$_f$	**0.113±0.001**	0.113±0.010	0.363±0.005	**0.172±0.003**
MLSI	0.221±0.051	0.052±0.030	0.342±0.007	0.296±0.017
CCA	0.115±0.000	0.003±0.001	0.344±0.005	0.288±0.005
PLS	0.115±0.000	0.001±0.000	0.344±0.005	0.230±0.015
PCA	0.115±0.001	0.079±0.005	0.358±0.002	0.193±0.005
LPP	0.114±0.000	0.078±0.005	0.364±0.005	0.225±0.005
SEL	0.115±0.001	0.085±0.004	0.362±0.004	0.210±0.002
ORI	0.114±0.000	0.025±0.004	0.349±0.006	0.224±0.003

Methods	OE ↓	CV($\times 10^{-3}$) ↓	AP ↑	AUC ↑
MDDM$_p$	0.666±0.016	**0.116± 0.003**	0.235±0.008	0.730±0.012
MDDM$_f$	**0.634±0.017**	**0.116± 0.003**	**0.275±0.007**	**0.733±0.012**
MLSI	0.945±0.104	0.179± 0.007	0.091±0.031	0.500±0.010
CCA	**0.637±0.025**	0.178± 0.003	0.271±0.008	0.610±0.008
PLS	0.866±0.033	0.139± 0.008	0.160±0.019	0.697±0.013
PCA	0.776±0.016	0.127± 0.004	0.221±0.008	0.679±0.006
LPP	0.806±0.013	0.142± 0.004	0.170±0.005	0.630±0.014
SEL	0.833±0.012	0.132± 0.003	0.167±0.006	0.616±0.007
ORI	0.729±0.022	0.144± 0.003	0.238±0.007	0.679±0.011

Table V. Results (mean±std.) on text categorization when ML-kNN is used as base classifier (↓ indicates "the smaller the better"; ↑ indicates "the larger the better". The best result and the results comparable with the best one are highlighted in boldface.)

Methods	HL($\times 10^1$) ↓	microF1 ↑	macroF1 ↑	RL ↓
MDDM$_p$	**0.227±0.007**	**0.778±0.005**	**0.627±0.008**	**0.035±0.001**
MDDM$_f$	0.265±0.007	0.752±0.003	0.607±0.010	0.063±0.003
MLSI	0.505±0.074	0.521±0.071	0.401±0.049	0.153±0.019
CCA	0.504±0.030	0.537±0.015	0.428±0.026	0.126±0.007
PLS	0.326±0.013	0.644±0.011	0.316±0.006	0.060±0.002
PCA	0.276±0.009	0.710±0.008	0.471±0.006	0.044±0.002
LPP	0.574±0.025	0.344±0.031	0.158±0.020	0.186±0.009
SEL	0.362±0.007	0.595±0.009	0.368±0.015	0.075±0.004
ORI	0.410±0.009	0.493±0.024	0.286±0.020	0.121±0.001
Methods	OE ↓	CV($\times 10^{-2}$) ↓	AP ↑	AUC ↑
MDDM$_p$	0**.065±0.006**	**0.068±0.002**	**0.874±0.003**	**0.899±0.004**
MDDM$_f$	0.089±0.010	0.103±0.004	0.828±0.004	0.820±0.003
MLSI	0.330±0.093	0.182±0.016	0.604±0.062	0.694±0.029
CCA	0.296±0.016	0.155±0.010	0.630±0.010	0.702±0.010
PLS	0.100±0.007	0.094±0.004	0.783±0.008	0.838±0.008
PCA	0.084±0.005	0.076±0.003	0.832±0.003	0.887±0.006
LPP	0.533±0.049	0.204±0.012	0.455±0.024	0.601±0.025
SEL	0.185±0.012	0.107±0.004	0.726±0.004	0.822±0.007
ORI	0.303±0.026	0.157±0.003	0.619±0.010	0.673±0.009

Table VI. Results (mean±std.) on text categorization when SVM is used as base classifier

Methods	HL ↓	microF1↑	macroF1 ↑	RL ↓
MDDM$_p$	**0.024±0.001**	**0.754±0.002**	**0.595±0.007**	**0.044±0.003**
MDDM$_f$	0.027±0.001	0.737±0.005	**0.594±0.011**	0.046±0.001
MLSI	0.110±0.033	0.388±0.064	0.355±0.043	0.179±0.036
CCA	0.109±0.006	0.387±0.017	0.383±0.016	0.207±0.014
PLS	0.031±0.001	0.653±0.006	0.321±0.012	0.071±0.002
PCA	0.029±0.001	0.673±0.006	0.374±0.014	0.052±0.003
LPP	0.051±0.001	0.215±0.031	0.117±0.017	0.358±0.032
SEL	0.037±0.001	0.561±0.011	0.304±0.024	0.107±0.009
ORI	**0.024±0.001**	0.738±0.004	**0.589±0.010**	0.046±0.001
Methods	OE ↓	CV($\times 10^{-2}$) ↓	AP ↑	AUC ↑
MDDM$_p$	**0.068±0.008**	**0.080±0.004**	**0.860±0.006**	**0.941±0.005**
MDDM$_f$	0.098±0.004	**0.080±0.001**	0.838±0.002	0.930±0.002
MLSI	0.495±0.091	0.201±0.031	0.507±0.073	0.818±0.025
CCA	0.584±0.076	0.230±0.018	0.448±0.032	0.812±0.011
PLS	0.083±0.007	0.105±0.004	0.783±0.003	0.906±0.004
PCA	0.081±0.006	0.084±0.004	0.815±0.004	0.923±0.002
LPP	0.581±0.067	0.314±0.018	0.383±0.039	0.606±0.029
SEL	0.179±0.006	0.136±0.007	0.694±0.012	0.860±0.010
ORI	0.073±0.003	0.083±0.001	0.838±0.001	0.934±0.002

Table VII. Results (mean±std.) of MDDM methods with/without embedding the label relationship on web page classification when ML-kNN is used as base classifier (↓ indicates "the smaller the better"; ↑ indicates "the larger the better". The best result and the results comparable with the best one are highlighted in boldface.)

Methods	HL($\times 10^1$) ↓	microF1 ↑	macroF1 ↑	RL($\times 10^1$) ↓
MDDM$_p^l$	0.419±0.128	0.465± 0.136	0.297± 0.069	0.953±0.358
MDDM$_p^k$	**0.415±0.129**	0.474± 0.134	0.301± 0.067	**0.949±0.358**
MDDM$_f^l$	0.453±0.137	**0.496± 0.113**	**0.325± 0.056**	0.989±0.368
MDDM$_f^k$	0.447±0.139	**0.498± 0.113**	**0.328± 0.057**	0.985±0.366
Methods	OE ↓	CV($\times 10^{-1}$) ↓	AP ↑	AUC ↑
MDDM$_p^l$	0.407±0.134	0.376± 0.118	0.672±0.102	0.651±0.022
MDDM$_p^k$	**0.403±0.134**	**0.374± 0.118**	**0.673±0.102**	**0.654±0.022**
MDDM$_f^l$	0.413±0.129	0.386± 0.121	0.665±0.100	0.649±0.025
MDDM$_f^k$	0.412±0.129	0.384± 0.121	0.666±0.100	0.651±0.025

Table VIII. Results (mean±std.) of MDDM methods with/without embedding the label relationship on web page classification when SVM is used as base classifier

Methods	HL($\times 10^1$) ↓	microF1 ↑	macroF1 ↑	RL ↓
MDDM$_p^l$	0.411±0.126	0.468±0.132	0.299±0.065	**0.147±0.045**
MDDM$_p^k$	**0.408±0.126**	0.468±0.135	0.295±0.068	**0.149±0.043**
MDDM$_f^l$	0.483±0.152	0.493±0.107	**0.333±0.053**	0.185±0.048
MDDM$_f^k$	0.463±0.143	**0.497±0.108**	**0.335±0.055**	0.185±0.051
Methods	OE ↓	CV($\times 10^{-1}$) ↓	AP ↑	AUC ↑
MDDM$_p^l$	0.399±0.128	**0.553±0.140**	**0.639±0.101**	**0.716±0.036**
MDDM$_p^k$	**0.396±0.126**	**0.560±0.137**	**0.640±0.098**	**0.724±0.035**
MDDM$_f^l$	0.419±0.123	0.694±0.199	0.619±0.098	0.695±0.033
MDDM$_f^k$	0.412±0.122	0.692±0.201	0.624±0.099	0.712±0.034

ranging from 1% to 100% of the dimension of the original space, with 2% as interval; on text categorization, since the original dimension is very high, to avoid useless costs, we run experiments with d ranging from 0.1% to 10% of the dimension of the original space, with 0.2% as interval.[4] Here we only present the results on *Hamming Loss* which is arguably the most important multi-label evaluation criterion.

It can be found from Fig. 2 and 3 that when ML-kNN is the base classifier, the performance of MDDM$_p$ is better than the compared methods on all the tasks under most d values. On image annotation, when SVM is the base classifier, the performance of MDDM$_p$ is worse than LPP and SEL when d is small, while MDDM$_p$ is much more stable than LPP and SEL. It is clear that MDDM$_p$ is superior to the compared methods for most cases.

We can also find that the performance curve on image annotation is not as smooth as that on the other two tasks. We conjecture that this is caused by the well-known large gap between low-level image features and high-level image semantics. In other words, the dependence between the feature description and the label information

[4] Here we randomly select from the orthonormal basis of the null space when $d > r$.

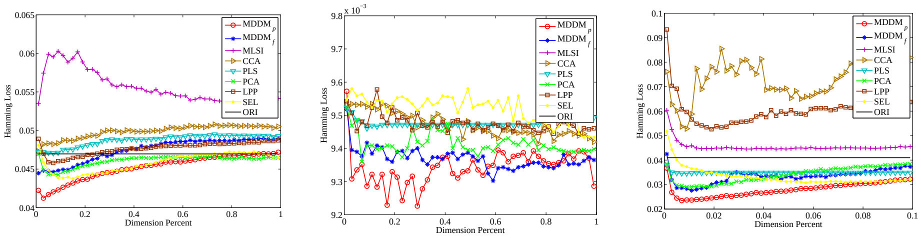
Table IX. Results (mean±std.) of MDDM methods with/without embedding the label relationship on image annotation when ML-kNN is used as base classifier (↓ indicates "the smaller the better"; ↑ indicates "the larger the better". The best result and the results comparable with the best one are highlighted in boldface.)

Methods	HL($\times 10^2$) ↓	microF1 ↑	macroF1 ↑	RL ↓
MDDM$_p^l$	**0.941±0.004**	0.160± 0.013	**0.484± 0.006**	**0.117± 0.002**
MDDM$_p^k$	**0.938±0.005**	0.189± 0.007	**0.485± 0.005**	**0.117± 0.002**
MDDM$_f^l$	**0.942±0.009**	**0.199± 0.009**	**0.486± 0.004**	0.122± 0.003
MDDM$_f^k$	**0.940±0.008**	**0.204± 0.007**	**0.487± 0.004**	0.121± 0.003
Methods	OE ↓	CV($\times 10^{-3}$) ↓	AP ↑	AUC ↑
MDDM$_p^l$	0.635± 0.008	**0.100±0.002**	**0.328± 0.007**	0.600±0.004
MDDM$_p^k$	**0.626± 0.012**	**0.100±0.002**	**0.330± 0.007**	**0.604±0.004**
MDDM$_f^l$	0.662± 0.009	0.103±0.003	0.300± 0.009	0.586±0.003
MDDM$_f^k$	0.659± 0.012	0.103±0.003	0.303± 0.009	0.590±0.004

Table X. Results (mean±std.) of MDDM methods with/without embedding the label relationship on image annotation when SVM is used as base classifier

Methods	HL($\times 10^2$) ↓	microF1 ↑	macroF1 ↑	RL ↓
MDDM$_p^l$	**0.941±0.005**	0.170±0.013	0.482±0.004	0.153±0.004
MDDM$_p^k$	**0.941±0.005**	0.172±0.012	**0.483±0.004**	**0.150±0.003**
MDDM$_f^l$	0.951±0.007	**0.191±0.009**	**0.484±0.005**	0.152±0.005
MDDM$_f^k$	0.950±0.006	**0.192±0.010**	**0.485±0.005**	**0.149±0.004**
Methods	OE ↓	CV($\times 10^{-3}$) ↓	AP ↑	AUC ↑
MDDM$_p^l$	0.660±0.018	0.127±0.003	0.250±0.008	0.730±0.012
MDDM$_p^k$	**0.659±0.017**	**0.124±0.003**	0.255±0.002	**0.735±0.013**
MDDM$_f^l$	0.662±0.016	0.127±0.003	0.255±0.006	**0.733±0.012**
MDDM$_f^k$	**0.657±0.012**	**0.123±0.003**	**0.262±0.004**	**0.737±0.009**

in the image annotation task is not as strong as that in the other two tasks.



(a) Web page classification (b) Image annotation (c) Text categorization

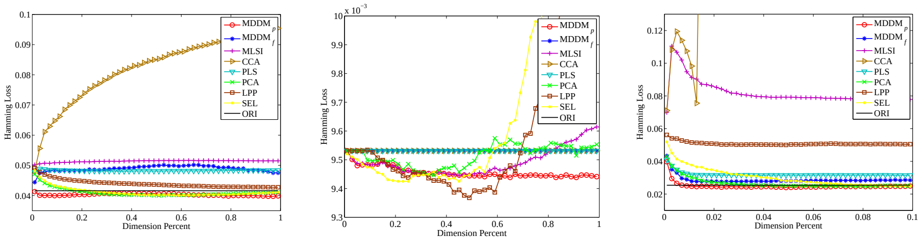
Fig. 2. Results with different dimensionalities of the lower-dimensional space when ML-kNN is used as base classifier

Table XI. Results (mean±std.) of MDDM methods with/without embedding the label relationship on text categorization when ML-kNN is used as base classifier (↓ indicates "the smaller the better"; ↑ indicates "the larger the better". The best result and the results comparable with the best one are highlighted in boldface.)

Methods	HL($\times 10^1$) ↓	microF1 ↑	macroF1 ↑	RL($\times 10^1$) ↓
MDDM^l_p	**0.227±0.007**	0.778±0.005	0.627±0.008	**0.354±0.009**
MDDM^k_p	**0.225±0.005**	**0.781±0.005**	**0.632±0.008**	**0.352±0.009**
MDDM^l_f	0.265±0.007	0.752±0.003	0.607±0.010	0.626±0.032
MDDM^k_f	0.263±0.009	0.753±0.004	0.607±0.009	0.622±0.036
Methods	OE($\times 10^1$) ↓	CV($\times 10^{-2}$) ↓	AP ↑	AUC ↑
MDDM^l_p	0.649±0.058	**0.068±0.002**	**0.874±0.003**	**0.899±0.004**
MDDM^k_p	**0.629±0.041**	**0.068±0.002**	**0.875±0.003**	**0.899±0.004**
MDDM^l_f	0.894±0.102	0.103±0.004	0.828±0.004	0.820±0.003
MDDM^k_f	0.869±0.095	0.103±0.004	0.829±0.004	0.822±0.004

Table XII. Results (mean±std.) of MDDM methods with/without embedding the label relationship on text categorization when SVM is used as base classifier

Methods	HL($\times 10^1$) ↓	microF1 ↑	macroF1 ↑	RL($\times 10^1$) ↓
MDDM^l_p	**0.242±0.005**	**0.754± 0.002**	**0.595± 0.007**	0.441±0.027
MDDM^k_p	**0.241±0.004**	**0.755± 0.002**	**0.596± 0.008**	**0.419±0.029**
MDDM^l_f	0.268±0.009	0.737± 0.005	**0.594± 0.011**	0.457±0.012
MDDM^k_f	0.267±0.008	0.738± 0.003	**0.595± 0.011**	0.450±0.008
Methods	OE($\times 10^1$) ↓	CV($\times 10^{-1}$) ↓	AP ↑	AUC ↑
MDDM^l_p	**0.677±0.076**	0.805±0.039	**0.860±0.006**	**0.941±0.005**
MDDM^k_p	**0.665±0.064**	**0.779±0.051**	**0.862±0.006**	**0.941±0.005**
MDDM^l_f	0.983±0.043	**0.803±0.007**	0.838±0.002	0.930±0.002
MDDM^k_f	0.967±0.038	**0.795±0.008**	0.839±0.001	0.930±0.002



(a) Web page classification (b) Image annotation (c) Text categorization

Fig. 3. Results with different dimensionalities of the lower-dimensional space when SVM is used as base classifier

5. CONCLUSION

Dimensionality reduction has been studied for many years, however, few results on multi-label dimensionality reduction have been reported. This paper extends our preliminary research [Zhang and Zhou 2008], which performs multi-label dimensionality reduction by maximizing the dependence between the feature description and the associated class labels. Experiments validate the performance of our proposed MDDM method.

Different from many other dimensionality reduction methods, MDDM provides a possibility of utilizing the relationship between labels to improve the performance via the label matrix **L** encoding the label correlation. Designing a better method for constructing **L** is an important future work. From the experiments we can see the superiority of MDDM is more apparent when the base classifier is ML-kNN. Since MDDM is designed independently to the base classifier, another important issue to be explored in the future is to design specific multi-label dimensionality reduction methods for SVM.

ACKNOWLEDGMENTS

We want to thank Kai Yu for providing the code of MLSI, Xin-Pan Xiao and Qing Wang for reading a preliminary draft, and the helpful comments and suggestions from the anonymous reviewers.

REFERENCES

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

Argyriou, A., Evgeniou, T., and Pontil, M. 2008. Convex multi-task feature learning. *Machine Learning 73,* 3, 243–272.

Bach, F. R. and Jordan, M. I. 2002. Kernel independent component analysis. *Journal of Machine Learning Research 3*, 1–48.

Bakker, B. and Heskes, T. 2003. Task clustering and gating for bayesian multictask learning. *Journal of Machine Learning Research 4*, 83–99.

Barutcuoglu, Z., Schapire, R. E., and Troyanskaya, O. G. 2006. Hierarchical multi-label prediction of gene function. *Bioinformatics 22,* 7, 830–836.

Belkin, M. and Niyogi, P. 2002. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. MIT Press, Cambridge, MA, 585–591.

Boutell, M. R., Luo, J., Shen, X., and Brown, C. M. 2004. Learning multi-label scene classification. *Pattern Recognition 37,* 9, 1757–1771.

Chen, J., Ji, S., Ceran, B., Li, Q., Wu, M., and Ye, J. 2008. Learning subspace kernels for classification. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Las Vegas, NV, 106–114.

Davis, J. V., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S. 2007. Information-theoretic metric learning. In *Proceedings of the the 24th International Conference on Machine Learning*. Corvallis, OR, 209–216.

Dumais, S. T., Platt, J., Heckerman, D., and Sahami, M. 1998. Inductive learning algorithms and representation for text categorization. In *Proceedings of the 7th ACM International Conference on Information and Knowledge Management*. Bethesda, MD, 148–155.

Duygulu, P., Barnard, K., de Freitas, N., and Forsyth, D. 2002. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceeding of the 7th European Conference on Computer Vision*. Copenhagen, Denmark, 97–112.

Elisseeff, A. and Weston, J. 2002. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. MIT Press, Cambridge, MA, 681–687.

Fisher, R. A. 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics 7,* 2, 179–188.

Ghamrawi, N. and McCallum, A. 2005. Collective multi-label classification. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*. Bremen, Germany, 195–200.

Gretton, A., Bousquet, O., Smola, A. J., and Schölkopf, B. 2005. Measuring statistical dependence with Hilbert-Schmidt norms. In *Proceedings of the 16th International Conference on Algorithmic Learning Theory*. Singapore, 63–77.

Hardoon, D. R., Szedmak, S., and Shawe-Taylor, J. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation 16,* 12, 2639–2664.

He, X. and Niyogi, P. 2004. Locality preserving projections. In *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. MIT Press, Cambridge, MA.

Joachims, T. 1998. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*. Chemnitz, Germany, 137–142.

Jolliffe, I. T. 1986. *Principal Component Analysis*. Springer, New York.

Kang, F., Jin, R., and Sukthankar, R. 2006. Correlated label propagation with application to multi-label learning. In *Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. New York, NY, 1719–1726.

Kazawa, H., Izumitani, T., Taira, H., and Maeda, E. 2005. Maximal margin labeling for multi-topic text categorization. In *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. MIT Press, Cambridge, MA, 649–656.

Lewis, D. D., Yang, Y., Rose, T., and Li, F. 2004. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research 5*, 361–397.

Liu, J., Ji, S., and Ye, J. 2009. Multi-task feature learning via efficient $l_{2,1}$-norm minimization. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. Montreal, Canada.

Liu, Y., Jin, R., and Yang, L. 2006. Semi-supervised multi-label learning by constrained nonnegative matrix factorization. In *Proceedings of the 21st National Conference on Artificial Intelligence*. Boston, MA, 421–426.

McCallum, A. 1999. Multi-label text classification with a mixture model trained by EM. In *Working Notes of AAAI'99 Workshop on Text Learning*. Orlando, FL.

Obozinski, G., Taskar, B., and Jordan, M. I. 2006. Multi-task feature selection. Tech. rep., Deptartment of Statistics, UC Berkeley.

Qi, G.-J., Hua, X.-S., Rui, Y., Tang, J., Mei, T., and Zhang, H.-J. 2007. Correlative multi-label video annotation. In *Proceedings of the 15th ACM International Conference on Multimedia*. Augsburg, Germany, 17–26.

Roweis, S. T. and Saul, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science 290,* 5500, 2323–2326.

Schapire, R. E. and Singer, Y. 2000. BoosTexter: A boosting-based system for text categorization. *Machine Learning 39,* 2-3, 135–168.

Song, L., Smola, A., Borgwardt, K., and Gretton, A. 2008. Colored maximum variance unfolding. In *Advances in Neural Information Processing Systems 20*, J. C. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. MIT Press, Cambridge, MA, 1385–1392.

Song, L., Smola, A., Gretton, A., Borgwardt, K., and Bedo, J. 2007. Supervised feature selection via dependence estimation. In *Proceeding of the 24th International Conference on Machine Learning*. Corvallis, OR, 823–830.

Sun, L., Ji, S., and Ye, J. 2008. Hypergraph spectral learning for multi-label classification. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Las Vegas, NV, 668–676.

Tenenbaum, J. B., de Silva, V., and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science 290,* 5500, 2319–2323.

Thabtah, F. A., Cowling, P. I., and Peng, Y. 2004. MMAC: A new multi-class, multi-label associative classification approach. In *Proceedings of the 4th IEEE International Conference on Data Mining*. Brighton, UK, 217–224.

Tikhonov, A. N. and Arsenin, V. Y. 1977. *Solutions of Ill-Posed Problems*. Wiley, New York.

UEDA, N. AND SAITO, K. 2003. Parametric mixture models for multi-labeled text. In *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. MIT Press, Cambridge, MA, 721–728.

WOLD, H. 1985. Partial least squares. *Encyclopedia of the Statistical Sciences 6*, 581–591.

YANG, Y. 1999. An evaluation of statistical approaches to text categorization. *Information Retrieval 1,* 1-2, 69–90.

YU, K., YU, S., AND TRESP, V. 2005. Multi-label informed latent semantic indexing. In *Proceedings of the 28th ACM SIGIR International Conference on Research and Development in Information Retrieval*. Salvador, Brazil, 258–265.

YU, S., YU, K., TRESP, V., AND KRIEGEL, H.-P. 2006. Multi-output regularized feature projection. *IEEE Transactions on Knowledge and Data Engineering 18,* 12, 1600–1613.

ZHANG, M.-L. AND ZHOU, Z.-H. 2006. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering 18,* 10, 1338–1351.

ZHANG, M.-L. AND ZHOU, Z.-H. 2007. ML-kNN: A lazy learning approach to multi-label learning. *Pattern Recognition 40,* 7, 2038–2048.

ZHANG, Y. AND ZHOU, Z.-H. 2008. Multi-label dimensionality reduction via dependency maximization. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*. Chicago, IL, 1503–1505.

ZHOU, Z.-H. AND ZHANG, M.-L. 2007. Multi-instance multi-label learning with application to scene classification. In *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hofmann, Eds. MIT Press, Cambridge, MA, 1609–1616.

ZHU, S., JI, X., XU, W., AND GONG, Y. 2005. Multi-labelled classification using maximum entropy method. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Salvador, Brazil, 274–281.

Received April 2009; revised September 2009; accepted February 2010