Extracting Shared Subspaces for Multi-label Classification

Shuiwang Ji Lei Tang
Arizona State University
Tempe, AZ 85287
shuiwang.ji@asu.edu
L.Tang@asu.edu

Shipeng Yu
Siemens Medical Solutions
Malvern, PA 19355
shipeng.yu@siemens.com

Jieping Ye
Arizona State University
Tempe, AZ 85287
jieping.ye@asu.edu

ABSTRACT

Multi-label problems arise in various domains such as multi-topic web page categorization, protein function prediction, and automated newswire classification. One natural way to deal with such problems is to construct a binary classifier for each label, resulting in a set of independent binary classification problems. Since the multiple labels share the same input space, and the semantics conveyed by different labels are usually correlated, it is essential to exploit the correlation information contained in different labels. In this paper, we consider a general framework for extracting shared structures in multi-label classification. In this framework, a common subspace is assumed to be shared among multiple labels. We show that the optimal solution to the proposed formulation can be obtained by solving a generalized eigenvalue problem, although the optimization problem is non-convex. For high-dimensional problems, direct computation of the solution is expensive, and we develop an efficient algorithm for this case. One appealing feature of the proposed framework is that it includes several well-known algorithms as special cases, thus elucidating their intrinsic relationships. We have conducted extensive experiments on eleven multi-topic web page categorization tasks, and results demonstrate the effectiveness of the proposed formulation in comparison with several representative algorithms.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications - Data Mining

General Terms

Algorithms

Keywords

Multi-label classification, shared subspace, least squares loss, singular value decomposition

1. INTRODUCTION

Learning and mining from objects annotated with multiple labels is a frequently encountered and widely studied problem in many domains. For example, in multi-topic web page categorization [27, 19, 28], a web page can be assigned to multiple topic categories. In gene and protein function prediction [6, 25], multiple functional labels are associated with each gene and protein, since an individual gene or protein usually performs multiple functions. In automated newswire categorization, multiple labels can be associated with a newswire story indicating its subject categories and the regional categories of reported events [34]. One common aspect of these problems, so-called multi-label classification, is that multiple labels are associated with a single object. Such problems are more general than the traditional multi-class problems in which a single label is assigned to an object. Driven by various applications, such problems are receiving increasing attention now [22, 16, 8, 33, 12, 18, 30].

One simple and popular approach for multi-label classification is to construct a binary classifier for each label in which instances relevant to this label form the positive class, and the rest form the negative class. This approach has been applied successfully to various applications [9, 31, 17, 9]. However, it fails to capture the correlation information among different labels, which is critical for many applications where the semantics conveyed by different labels are correlated. Indeed, it has been shown that the decoupling of multiple labels may compromise the performance significantly in certain applications [27]. For example, in modeling the topics and authorship of documents, it is evident that the topics and authors of documents are correlated, since a particular author may only write on certain topics. Hence, it is essential to model them in a coordinated fashion so that their intrinsic relationships can be captured.

In this paper, we propose a general framework for extracting shared structures (subspace) in multi-label classification. In this framework, a binary classifier is constructed for each label to discriminate this label from the rest of them. However, unlike the approach that build the binary classifier independently, a low-dimensional subspace is assumed to be shared among multiple labels. The predictive functions in our formulation consist of two parts: the first part is contributed from the representations in the original data space, and the second one is contributed from the embedding in the shared subspace. A similar formulation has been proposed in [3] for multi-task learning. We show that when the least squares loss is used in classification, the linear transformation that characterizes the shared subspace can be computed by solving a generalized eigenvalue problem. In contrast, the formulation proposed in [3] is non-convex and needs to be solved iteratively. For high-dimensional problems, direct computation of the solution is computationally expensive, and we develop an efficient algorithm for this case. One appealing feature of the proposed framework is that it includes several well-known algorithms as special cases, thus elucidating their intrinsic relationships. We have conducted extensive experiments on eleven multi-topic web page categorization tasks, and results demonstrate the effectiveness of the proposed formulation. Experimental results also show that the proposed formulation based on least squares loss is comparable to other formulations based on hinge loss, while it is much more efficient.

The key contributions of this paper are highlighted as follows:

- We propose a general framework for extracting shared structures in multi-label classification. In this framework, the correlation information among multiple labels is captured by a low-dimensional subspace shared among all labels.
- We show that when the least squares loss is used in classification, the shared structure can be computed by solving a generalized eigenvalue problem. To reduce the computational cost, we propose an efficient algorithm for high-dimensional problems.
- We show that the proposed formulation is a general one that includes several well-known formulations as special cases.
- We have conducted extensive experiments on eleven multi-topic web page categorization tasks to demonstrate the effectiveness of the proposed formulation.

The rest of this paper is organized as follows: We present the framework for extracting shared subspace in Section 2. The efficient algorithm for computing the solution is developed in Section 3. We discuss its relationship with existing formulations in Section 4 and report experimental results in Section 5. This paper includes with conclusions and discussions in Section 6.

Notations: We use n, d, and m to denote the number of training instances, the data dimensionality, and the number of labels, respectively. The data matrix and the label indicator matrix are denoted as $X = [x_1, \cdots, x_n]^T \in \mathbb{R}^{n\times d}$ and $Y \in \mathbb{R}^{n\times m}$, where $x_i \in \mathbb{R}^d$ is the ith instance, and $Y_{i\ell} = 1$ if the ith instance has the ℓth label, and -1 otherwise.

2. THE PROPOSED FRAMEWORK

We are given a set of input data $\{x_i\}_{i=1}^n \in \mathbb{R}^d$ and the class label indicator matrix $Y \in \mathbb{R}^{n\times m}$ that encodes the label information, where m and n are the number of labels and the number of instances, respectively. Following the traditional supervised learning framework, we learn m functions $\{f_\ell\}_{\ell=1}^m$ from the data that minimize the following regularized empirical risk:

$$R(\{f_\ell\}_{\ell=1}^m) = \sum_{\ell=1}^m \left(\sum_{i=1}^n L(f_\ell(x_i), y_i^\ell) + \alpha \Omega(f_\ell) \right), \qquad (1)$$

where $y_i^\ell = Y_{i\ell}$, L is a prescribed loss function, $\Omega(f)$ is a regularization functional measuring the smoothness of f, and $\alpha > 0$ is the regularization parameter.

2.1 Problem Formulation

We propose a multi-label learning framework, in which a low-dimensional subspace is shared by all labels. The predictive functions in this framework consist of two parts: one part is contributed from the original data space, and another part is derived from the shared subspace as follows:

$$f_\ell(x) = \mathbf{w}_\ell^T x + \mathbf{v}_\ell^T \Theta x, \qquad (2)$$

where $\mathbf{w}_\ell \in \mathbb{R}^d$ and $\mathbf{v}_\ell \in \mathbb{R}^r$ are the weight vectors, $\Theta \in \mathbb{R}^{r\times d}$ is the linear transformation used to parameterize the shared low-dimensional subspace, and r is the dimensionality of the shared subspace. The transformation Θ is common for all labels, and it has orthonormal rows, that is $\Theta\Theta^T = I$. In this formulation, the input data are projected onto a low-dimensional subspace by Θ, and this low-dimensional projection is combined with the original representation to produce the final prediction. Note that a similar formulation has been proposed in [3] to capture the shared predictive structures in multi-task learning, and our formulation differs with it in important aspects (see Section 4.1 for a comparison).

Following the regularization formulation in Eq. (1), we propose to estimate the parameters $\{\mathbf{w}_\ell, \mathbf{v}_\ell\}_{\ell=1}^m$ and Θ by minimizing the following regularized empirical risk:

$$\sum_{\ell=1}^m \left(\frac{1}{n}\sum_{i=1}^n L((\mathbf{w}_\ell + \Theta^T\mathbf{v}_\ell)^T x_i, y_i^\ell) + \alpha||\mathbf{w}_\ell||^2 + \beta||\mathbf{w}_\ell + \Theta^T\mathbf{v}_\ell||^2\right),$$

subject to the constraint that $\Theta\Theta^T = I$. Note that in the above formulation, the first regularization term $||\mathbf{w}_\ell||^2$ controls the amount of information shared by all labels, while the second regularization term $||\mathbf{w}_\ell + \Theta^T\mathbf{v}_\ell||^2$ controls the complexity of the model. By a change of variable, this problem can be reformulated equivalently as follows:

$$\min_{\{\mathbf{u}_\ell, \mathbf{v}_\ell\}, \Theta} \sum_{\ell=1}^m \left(\frac{1}{n}\sum_{i=1}^n L(\mathbf{u}_\ell^T x_i, y_i^\ell) + \alpha||\mathbf{u}_\ell - \Theta^T\mathbf{v}_\ell||^2 + \beta||\mathbf{u}_\ell||^2 \right)$$
$$\text{s. t.} \quad \Theta\Theta^T = I. \qquad (3)$$

In this paper, we consider the least squares loss, i.e.,

$$L(\mathbf{u}_\ell^T x_i, y_i^\ell) = (\mathbf{u}_\ell^T x_i - y_i^\ell)^2.$$

It has been shown [11, 24] that the least squares loss function is comparable to other loss functions such as the hinge loss employed in support vector machines (SVM) [26] when appropriate regularization is added. Hence, we get the following optimization problem:

$$\min_{\{\mathbf{u}_\ell, \mathbf{v}_\ell\}, \Theta} \sum_{\ell=1}^m \left(\frac{1}{n}||X\mathbf{u}_\ell - y_\ell||^2 + \alpha||\mathbf{u}_\ell - \Theta^T\mathbf{v}_\ell||^2 + \beta||\mathbf{u}_\ell||^2 \right)$$
$$\text{s. t.} \quad \Theta\Theta^T = I, \qquad (4)$$

where $X = [x_1, \cdots, x_n]^T \in \mathbb{R}^{n\times d}$ is the data matrix, $y_\ell = [y_1^\ell, \cdots, y_n^\ell]^T \in \mathbb{R}^n$. The formulation in Eq. (4) can be expressed compactly as:

$$\min_{U,V,\Theta} \quad \frac{1}{n}||XU - Y||_F^2 + \alpha||U - \Theta^T V||_F^2 + \beta||U||_F^2$$
$$\text{s. t.} \quad \Theta\Theta^T = I, \qquad (5)$$

where $\|\cdot\|_F$ denotes the Frobenius norm of matrix [13], $U = [\mathbf{u}_1, \cdots, \mathbf{u}_m]$, and $V = [\mathbf{v}_1, \cdots, \mathbf{v}_m]$.

2.2 The Computation of V^*

We show that the optimal V^* that solves the optimization problem in Eq. (5) can be expressed in terms of Θ and U, as summarized in the following lemma:

Lemma 2.1. *Let U, V, and Θ be defined as above. Then the optimal V^* that solves the optimization problem in Eq. (5) is given by $V^* = \Theta U$.*

Proof. The only term in Eq. (5) that depends on V is $||U - \Theta^T V||_F^2$, which can be expressed equivalently as:

$$\begin{aligned} ||U - \Theta^T V||_F^2 &= \mathrm{tr}(U^T - V^T\Theta)(U - \Theta^T V) \\ &= \mathrm{tr}(U^T U + V^T \Theta\Theta^T V - 2U^T\Theta^T V), \end{aligned} \tag{6}$$

where $\mathrm{tr}(\cdot)$ denote the trace of matrix, and we have used the property that $||A||_F^2 = \mathrm{tr}(A^T A)$ for any matrix A. Taking the derivative of the expression in Eq. (6) with respect to V, and setting it to zero, we obtain

$$V^* = \Theta U,$$

where we have used the property that $\Theta\Theta^T = I$. This completes the proof of the lemma. □

2.3 The Computation of U^*

It follows from Lemma 2.1 that the objective function in Eq. (5) can be rewritten as:

$$\begin{aligned} & \frac{1}{n}\|XU - Y\|_F^2 + \alpha||U - \Theta^T V||_F^2 + \beta||U||_F^2 \\ = \; & \frac{1}{n}\|XU - Y\|_F^2 + \alpha||U - \Theta^T \Theta U||_F^2 + \beta||U||_F^2 \\ = \; & \frac{1}{n}\|XU - Y\|_F^2 + \mathrm{tr}\left(U^T\left((\alpha+\beta)I - \alpha\Theta^T\Theta\right)U\right). \end{aligned} \tag{7}$$

Hence, the optimization problem in Eq. (5) can be expressed equivalently as:

$$\begin{aligned} \min_{U,\Theta} \quad & \frac{1}{n}\|XU - Y\|_F^2 + \mathrm{tr}\left(U^T\left((\alpha+\beta)I - \alpha\Theta^T\Theta\right)U\right) \\ \text{s. t.} \quad & \Theta\Theta^T = I. \end{aligned} \tag{8}$$

We show that the optimal U^* can be expressed in terms of Θ. This is summarized in the following lemma:

Lemma 2.2. *Let X, Y, U, and Θ be defined as above. Then the optimal U^* that solves the optimization problem in Eq. (8) can be expressed as:*

$$U^* = \frac{1}{n}\left(M - \alpha\Theta^T\Theta\right)^{-1} X^T Y, \tag{9}$$

where M is defined as:

$$M = \frac{1}{n}X^T X + (\alpha+\beta)I. \tag{10}$$

Proof. Taking the derivative of the objective function in Eq. (8) with respect to U, and setting it to zero, we obtain

$$U^* = \frac{1}{n}\left(M - \alpha\Theta^T\Theta\right)^{-1} X^T Y, \tag{11}$$

where M is defined in Eq. (10). □

2.4 The Computation of Θ^*

It follows from Lemma 2.2 that we can substitute the expression for U^* in Eq. (9) into Eq. (8) and obtain the following optimization problem with respect to Θ:

$$\begin{aligned} \max_{\Theta} \quad & \frac{1}{n^2}\mathrm{tr}\left(Y^T X \left(M - \alpha\Theta^T\Theta\right)^{-1} X^T Y\right) \\ \text{s. t.} \quad & \Theta\Theta^T = I. \end{aligned} \tag{12}$$

We show in the following theorem that the optimization problem in Eq. (12) can be simplified, and the optimal Θ^* can be obtained by solving a generalized eigenvalue problem.

Theorem 2.1. *Let X, Y, and Θ be defined as above. Then the optimal Θ^* that solves the optimization problem in Eq. (12) can be obtained by solving the following trace maximization problem:*

$$\begin{aligned} \max_{\Theta} \quad & tr\left(\left(\Theta S_1 \Theta^T\right)^{-1} \Theta S_2 \Theta^T\right) \\ s.\ t. \quad & \Theta\Theta^T = I, \end{aligned} \tag{13}$$

where S_1 and S_2 are defined as:

$$S_1 = I - \alpha M^{-1}, \tag{14}$$

$$S_2 = M^{-1} X^T Y Y^T X M^{-1}, \tag{15}$$

and M is defined in Eq. (10).

Proof. We need the Sherman-Woodbury-Morrison formula [13] for computing matrix inverse:

$$(P + ST)^{-1} = P^{-1} - P^{-1}S\left(I + TP^{-1}S\right)^{-1}TP^{-1}. \tag{16}$$

It follows from the formula in Eq. (16) that

$$\begin{aligned} & \left(M - \alpha\Theta^T\Theta\right)^{-1} \\ = \; & M^{-1} + \alpha M^{-1}\Theta^T\left(I - \alpha\Theta M^{-1}\Theta^T\right)^{-1}\Theta M^{-1} \\ = \; & M^{-1} + \alpha M^{-1}\Theta^T\left(\Theta\left(I - \alpha M^{-1}\right)\Theta^T\right)^{-1}\Theta M^{-1}, \end{aligned} \tag{17}$$

where the last equality follows since $\Theta\Theta^T = I$. By substituting the expression in Eq. (17) into the optimization problem in Eq. (12), we obtain the following problem:

$$\begin{aligned} & \max_{\Theta}\ \mathrm{tr}\left(Y^T X M^{-1}\Theta^T\left(\Theta\left(I - \alpha M^{-1}\right)\Theta^T\right)^{-1}\Theta M^{-1}X^T Y\right) \\ & \text{s. t. } \Theta\Theta^T = I, \end{aligned} \tag{18}$$

where we have omitted the term $Y^T X M^{-1} X^T Y$ since it is independent of Θ. By using the property that $\mathrm{tr}(AB) = \mathrm{tr}(BA)$ for any two matrices A and B, and noticing the definitions of S_1 and S_2 in Eqs. (14) and (15), respectively, we prove this theorem. □

Let $Z = [z_1, \cdots, z_r]$ be the matrix consisting of the top r eigenvectors corresponding to the largest r nonzero eigenvalues of the generalized eigenvalue problem: $S_1 z = \lambda S_2 z$. Let $Z = Z_q Z_r$ be the QR decomposition of Z, where Z_q has orthonormal columns and Z_r is upper triangular. It is easy to verify [32] that the objective function in Eq. (13) is invariant of any nonsingular transformation, that is, Q and NQ

achieve the same objective value for any nonsingular matrix $N \in \mathbb{R}^{r\times r}$. It follows that the optimal Q^* solving Eq. (13) is given by $Q^* = Z_q^T$. Note that S_1 is positive definite (see Eq. (20) below), thus Z can also be obtained by computing the top eigenvectors of $S_1^{-1}S_2$.

3. AN EFFICIENT ALGORITHM

From the discussions in the last section, the optimal Θ^* is given by the eigenvectors of $S_1^{-1}S_2 \in \mathbb{R}^{d\times d}$ corresponding to the r largest eigenvalues. When the data dimensionality, i.e., d, is small, the eigenvectors of $S_1^{-1}S_2$ can be computed directly. However, when d is large, direct eigendecomposition is computationally expensive. In this section, we show how we can compute the eigenvectors efficiently for this case.

3.1 Reformulation of $S_1^{-1}S_2$

It follows from the Sherman-Woodbury-Morrison formula in Eq. (16) that

$$\begin{aligned} M^{-1} &= \frac{1}{\alpha+\beta}I - \frac{1}{n(\alpha+\beta)^2}X^T\left(I + \frac{1}{n(\alpha+\beta)}XX^T\right)^{-1}X \\ &= \frac{1}{\alpha+\beta}I - \frac{1}{\alpha+\beta}X^T\left(n(\alpha+\beta)I + XX^T\right)^{-1}X. \quad (19) \end{aligned}$$

Hence, we have

$$I - \alpha M^{-1} = \frac{\beta}{\alpha+\beta}I + \frac{\alpha}{\alpha+\beta}X^T\left(n(\alpha+\beta)I + XX^T\right)^{-1}X, \tag{20}$$

which is positive definite when $\beta > 0$.

It follows from the definitions of M, S_1, and S_2 in Eqs. (10), (14), and (15) that

$$\begin{aligned} & S_1^{-1}S_2 \\ &= \left(I - \alpha M^{-1}\right)^{-1} M^{-1}X^TYY^TXM^{-1} \\ &= (M - \alpha I)^{-1}X^TYY^TXM^{-1} \quad (21) \\ &= \left(\frac{1}{n}X^TX + \beta I\right)^{-1} X^TYY^TX\left(\frac{1}{n}X^TX + (\alpha+\beta)I\right)^{-1}. \end{aligned}$$

Let

$$X = U\Sigma V^T \tag{22}$$

be the singular value decomposition (SVD) [13] of X, where $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{d\times d}$ are orthogonal, $\Sigma = \text{diag}(\Sigma_t, 0) \in \mathbb{R}^{n\times d}$ is diagonal, and $t = \text{rank}(X)$. Let $U = [U_1, U_2]$, where $U_1 \in \mathbb{R}^{n\times t}$ and $U_2 \in \mathbb{R}^{n\times(n-t)}$, $V = [V_1, V_2]$, where $V_1 \in \mathbb{R}^{d\times t}$ and $V_2 \in \mathbb{R}^{d\times(d-t)}$, and Σ_t consists of the first t rows and the first t columns of Σ. Then we have

$$\begin{aligned} & S_1^{-1}S_2 \\ =& V_1(\frac{1}{n}\Sigma_t^2 + \beta I)^{-1}V_1^TX^TYY^TX\left(\frac{1}{n}X^TX + (\alpha+\beta)I\right)^{-1} \\ & + \frac{1}{\beta}V_2I^{-1}V_2^TX^TYY^TX\left(\frac{1}{n}X^TX + (\alpha+\beta)I\right)^{-1} \\ =& V_1(\frac{1}{n}\Sigma_t^2 + \beta I)^{-1}V_1^TX^TYY^TX\left(\frac{1}{n}X^TX + (\alpha+\beta)I\right)^{-1} \\ =& V_1(\frac{1}{n}\Sigma_t^2 + \beta I)^{-1}V_1^TX^TYY^TXV_1(\frac{1}{n}\Sigma_t^2 + (\alpha+\beta)I)^{-1}V_1^T \\ =& V_1(\frac{1}{n}\Sigma_t^2 + \beta I)^{-1}\Sigma_tU_1^TYY^TU_1\Sigma_t(\frac{1}{n}\Sigma_t^2 + (\alpha+\beta)I)^{-1}V_1^T. \end{aligned}$$

The second and the third equalities follow since the columns of V_2 are in the null space of X, that is, $XV_2 = 0$.

3.2 Diagonalization of $S_1^{-1}S_2$

Define three diagonal matrices D_1, D_2, and D as follows:

$$D_1 = (\frac{1}{n}\Sigma_t^2 + \beta I)^{-1}\Sigma_t \in \mathbb{R}^{t\times t}, \tag{23}$$

$$D_2 = \Sigma_t(\frac{1}{n}\Sigma_t^2 + (\alpha+\beta)I)^{-1} \in \mathbb{R}^{t\times t}, \tag{24}$$

$$D = (D_1D_2^{-1})^{\frac{1}{2}} \in \mathbb{R}^{t\times t}. \tag{25}$$

Then we have

$$\begin{aligned} S_1^{-1}S_2 &= V_1D_1U_1^TYY^TU_1D_2V_1^T \\ &= V_1D(D^{-1}D_1)U_1^TYY^TU_1(D_2D)D^{-1}V_1^T \\ &= V_1D\tilde{D}U_1^TYY^TU_1\tilde{D}D^{-1}V_1^T, \end{aligned}$$

where

$$\tilde{D} = D^{-1}D_1 = D_2D. \tag{26}$$

Denote $C = Y^TU_1\tilde{D} \in \mathbb{R}^{m\times t}$ and let

$$C = P_1\Lambda P_2^T \tag{27}$$

be the SVD of C where $P_1 \in \mathbb{R}^{m\times m}$ and $P_2 \in \mathbb{R}^{t\times t}$ are orthogonal, and $\Lambda \in \mathbb{R}^{m\times t}$ is diagonal. Then

$$\begin{aligned} S_1^{-1}S_2 &= V_1DP_2\Lambda^T\Lambda P_2^TD^{-1}V_1^T \\ &= V_1DP_2\tilde{\Lambda}P_2^TD^{-1}V_1^T, \quad (28) \end{aligned}$$

where $\tilde{\Lambda} = \Lambda^T\Lambda \in \mathbb{R}^{t\times t}$.

3.3 Algorithms for Computing Θ^* and U^*

It follows from Eq. (28) that the eigenvectors of $S_1^{-1}S_2$ corresponding to nonzero eigenvalues are given by the columns of V_1DP_2. The algorithm for computing the optimal Θ^* for high-dimensional data is summarized as follows:

- Compute the SVD of X as $X = U\Sigma V^T = U_1\Sigma_tV_1^T$.
- Compute D_1, D_2, D, and $\tilde{D}$ as in Eqs. (23), (24), (25) and (26), respectively.
- Compute the SVD of $C = Y^TU_1\tilde{D}$ as $C = P_1\Lambda P_2^T$.
- Compute the QR decomposition of V_1DP_2 as $V_1DP_2 = QR$.
- The rows of the optimal Θ^* are given by the first r columns of the matrix Q.

After obtaining Θ^*, we need to compute the optimal U^* given by Eq. (9). Note that the matrix $M \in \mathbb{R}^{d\times d}$ is involved in Eq. (9), and hence it is expensive to compute U^* directly for high-dimensional data. More specifically, we need to make use of the expressions in Eqs. (17), (19), and (20) so that explicit formations of the matrices M and M^{-1} are avoided.

The SVD of X in the first step takes $O(dn^2)$ time assuming $d > n$. The size of C is $m \times t$ where m is the number of tasks and $t = \text{rank}(X)$. Hence the SVD of C in the third step takes $O(tm^2)$ time assuming $t > m$. The QR decomposition

Table 1: Summary of relevant matrices. The size, computation required, and the associated complexity of each relevant matrix are listed.

Matrix	Size	Computation	Complexity
X	$n \times d$	SVD	$O(dn^2)$
C	$m \times t$	SVD	$O(tm^2)$
$V_1 D P_2$	$d \times t$	QR	$O(dt^2)$

in the fourth step takes $O(dt^2)$ time. Typically, m and t are both small. Thus, the cost of the proposed algorithm for computing Θ^* is dominated by the cost for computing the SVD of X. A summary of relevant matrices and their associated computational complexity are listed in Table 1.

4. RELATIONSHIP TO EXISTING ALGORITHMS

In this section, we show that the proposed formulation includes several well-known algorithms as special cases. We begin by discussing related work.

4.1 Related Work

Dimensionality Reduction Canonical correlation analysis (CCA) [15] and partial least squares (PLS) [29, 4] are classical techniques for modeling relations between sets of observed variables. They both compute low-dimensional embedding of sets of variables simultaneously. Their main difference is that CCA maximizes the correlations between variables in the embedded space, while PLS maximizes their covariances. One popular use of CCA and PLS is for supervised learning, in which one set of variables are derived from the data and another set is derived from the class labels. In this setting, the data can be projected onto a lower-dimensional space directed by the label information. Such formulation is particularly appealing in the context of dimensionality reduction for multi-label data. When applied to multi-class problems, CCA reduces to the well-known linear discriminant analysis (LDA) formulation [10] in which a projection is obtained by maximizing the ratio of inter-class distance to intra-class distance.

Multi-task learning In [3], a similar formulation has been proposed for multi-task learning. In this formulation, the input data for different tasks can be different, and the following optimization problem is involved:

$$\begin{aligned} \min_{\{\mathbf{u}_\ell, \mathbf{v}_\ell\}, \Theta} \quad & \sum_{\ell=1}^{m} \left(\frac{1}{n_\ell} \sum_{i=1}^{n_\ell} L\left(\mathbf{u}_\ell^T x_i^\ell, y_i^\ell\right) + \alpha ||\mathbf{u}_\ell - \Theta^T \mathbf{v}_\ell||^2 \right) \\ \text{s. t.} \quad & \Theta\Theta^T = I, \end{aligned} \tag{29}$$

where x_i^ℓ is the ith instance in the ℓth task and n_ℓ is the number of instances in the ℓth task. It is shown [3] that the resulting optimization problem is non-convex even for convex loss functions. Hence, an iterative procedure called the *alternating structure optimization* (ASO) algorithm is proposed to compute a locally optimal solution. A similar idea of sharing part of the model parameters among multiple tasks has been explored in the Bayesian framework [5].

Multi-class learning Formulation for extracting shared structures in multi-class classification has been proposed recently [1]. In this formulation, a low-rank transformation is computed to uncover the shared structures in multi-class classification. The final prediction is solely based on the low-dimensional representations in the dimensionality-reduced space. Moreover, the low-rank constraint is non-convex, and it is first relaxed to the convex trace norm constraint. The relaxed problem can be formulated as a semidefinite program which is expensive to solve. Hence, gradient-based optimization technique is employed to solve the relaxed problem.

4.2 Connections with Existing Formulations

The formulation proposed in Section 2 includes several existing algorithm as special cases. In particular, by setting the regularization parameters α and β in Eq. (5) to different values, we obtain several well-known algorithms.

- $\alpha = 0$: When the regularization parameter $\alpha = 0$, it can be seen from Eq. (5) that this formulation is equivalent to the classical ridge regression [14]. In ridge regression, the multiple labels are decoupled, and the solution to each label can be obtained independently by solving a system of linear equations. In this case, no shared information is exploited among different labels.

- $\beta = 0$: When the regularization parameter $\beta = 0$, only the task-specific parameters $\{\mathbf{w}_\ell\}_{\ell=1}^m$ are regularized. Thus, the proposed formulation reduces to the one in [3] in the special case where the input data are the same for all tasks.

- $\alpha = +\infty$: It can be seen from Eq. (21) that when α tends to infinity,

  $$\left(\frac{1}{n} X^T X + (\alpha + \beta) I\right)^{-1} \to \epsilon I,$$

  for some small positive ϵ. Hence, the eigenvectors of $S_1^{-1} S_2$ is equivalent to the eigenvectors of the matrix

  $$\left(\frac{1}{n} X^T X + \beta I\right)^{-1} X^T Y Y^T X. \tag{30}$$

  This formulation is the same as the problem solved by orthonormalized PLS [4]. When the matrix YY^T in Eq. (30) is replaced by $Y(Y^TY)^{-1}Y^T$, this problem reduces to CCA. In the special case of multi-class problems, where each data point belongs to one class only, we define the class indicator matrix Y as follows: $y_{ij} = \sqrt{n/n_j} - \sqrt{n_j/n}$ if $y_i = j$, and $-\sqrt{n_j/n}$ otherwise, where n_j is the sample size of the j-th class. It is easy to verify that $\frac{1}{n} X^T X$ and $X^T Y Y^T X$ correspond to the total scatter and inter-class scatter matrices used in LDA. Thus, the optimal Θ coincides with the optimal transformation computed by LDA.

- $\beta = +\infty$: When β tends to infinity, the eigenvectors of $S_1^{-1} S_2$ is given by the eigenvectors of the matrix $X^T Y Y^T X$, which is the inter-class scatter matrix used in LDA. In this case, the proposed formulation is closely related to the orthogonal centroid method (OCM) [23] in which the optimal transformation is given by the eigenvectors of the inter-class scatter matrix corresponding to the largest eigenvalues.

5. EXPERIMENTAL STUDY

In this section, we evaluate the proposed formulation on multi-topic web page categorization tasks.

5.1 Experimental Setup

The multi-topic web page categorization data sets were described in [27, 19, 28], and they were compiled from 11 top-level categories in the "yahoo.com" domain. The web pages collected from each top-level category form a data set. The top-level categories are further divided into a number of second-level subcategories, and those subcategories form the topics to be categorized in each data set. Note that the 11 multi-topic categorization problems are compiled and solved independently as in [27]. We preprocess the data sets by removing topics with less than 100 web pages, words occurring less than 5 times, and web pages without topics. We use the TF-IDF encoding to represent web pages, and all web pages are normalized to unit length. The statistics of all data sets are summarized in Table 2.

Table 2: Statistics of the Yahoo data sets. m, d, and N denote the number of labels, the data dimensionality, and the total number of instance, respectively, in the data set after preprocessing. "MaxNPI"/"MinNPI" denotes the maximum/minimum number of positive instances for each topic (label).

Data set	m	d	N	MaxNPI	MinNPI
Arts	19	17973	7441	1838	104
Business	17	16621	9968	8648	110
Computers	23	25259	12371	6559	108
Education	14	20782	11817	3738	127
Entertainment	14	27435	12691	3687	221
Health	14	18430	9109	4703	114
Recreation	18	25095	12797	2534	169
Reference	15	26397	7929	3782	156
Science	22	24002	6345	1548	102
Social	21	32492	11914	5148	104
Society	21	29189	14507	7193	113

We use area under the receiver operating characteristic (ROC) curve, called AUC, and F1 score as the performance measure. To measure the performance across multiple labels using F1 score, we use both the macro F1 and the micro F1 scores [21, 31]. The F1 score depends on the threshold values of the classification models, and the thresholds computed by models are usually not optimized for it. Indeed, all methods yield low F1 scores when the thresholds computed by models are used. It is shown recently [9] that tuning the threshold based on F1 score on the training data can significantly improve performance. Hence, we tune the threshold value of each model based on the training data.

5.2 Performance Evaluation

We evaluate the proposed formulation on the 11 multi-topic web page categorization data sets. The experimental results on five relevant methods are also reported. Parameters of all the methods are tuned using 5-fold cross-validation based on F1 score. The setup is summarized as follows:

- **ML$_{\mathbf{LS}}$:** The proposed multi-label formulation. The regularization parameters α and β are tuned using 5-fold double cross-validation from the candidate set $[0, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1]$. The performance of the proposed formulation is not sensitive to the dimensionality of the shared subspace r as long as it is not too small. Hence, it is fixed to $5 \times \lfloor (m-1)/5 \rfloor$ in the experiments where m is the number of labels.

- **CCA+Ridge:** CCA is applied first to reduce the data dimensionality before ridge regression is applied. The regularization parameters for CCA and ridge regression are tuned on the set $\{10^i | i = -6, -5, \cdots, 1\}$ and $\{10^i | i = -6, -5, \cdots, 0\}$, respectively.

- **CCA+SVM:** CCA is applied first to reduce the data dimensionality before linear SVM is applied. The regularization parameter for CCA is tuned on the set $\{10^i | i = -6, -5, \cdots, 1\}$, and the C value for SVM is tuned on the set $\{10^i | i = -4, -3, \cdots, 4, 5\}$.

- **ASO$_{\mathbf{SVM}}$:** The alternating structural optimization (ASO) algorithm proposed in [3] with hinge loss as described in Eq. (29). The regularization parameter α is tuned on the set $\{10^i | i = -4, -3, \cdots, 2, 3\}$. The tolerance parameter for testing convergence is set to 10^{-3}, and the maximum number of iterations for ASO is set to 100. The optimization problem involved is solved using the MOSEK package [2].

- **SVM:** Linear SVM is applied on each label using the one-against-rest scheme, and the C value for each SVM is tuned on the set $\{10^i | i = -5, -4, \cdots, 4, 5\}$.

- **SVM$_{\mathbf{C}}$:** Linear SVM is applied on each label using the one-against-rest scheme, and $C = 1$ for all SVMs.

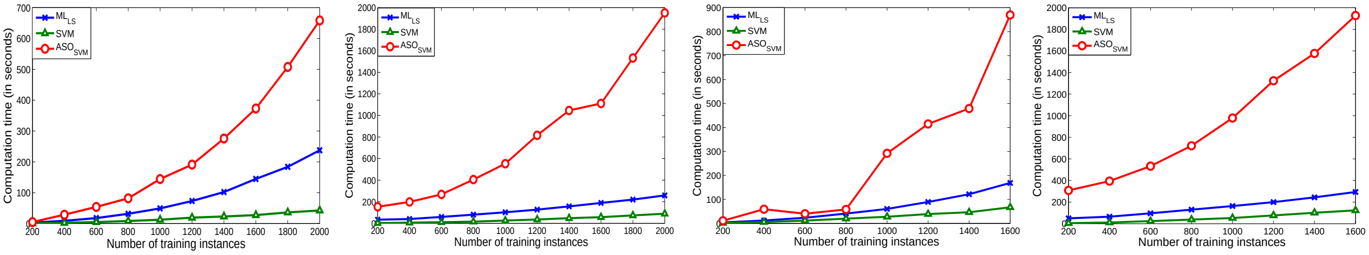
The SVM problems are solved using the LIBSVM [7] software package. All the codes and data sets used for the experiments are available at the supplemental website[1].

We randomly sample 1000 data points from each data set as training data (each label is guaranteed to appear in at least one data point), and the remaining data points are used as test data. This process is repeated five times to generate five random training/test partitions, and the averaged performance and standard deviations are reported. Tables 3 and 4 show the performance of the six methods in terms of AUC, macro F1, and micro F1. We can observe that the proposed formulation outperforms all other five compared methods in terms of macro F1 score on all of the 11 data sets. In terms of AUC, the proposed formulation achieves the highest AUC on 9 data sets, while SVM-based methods achieve the highest AUC on the other two data sets. In terms of micro F1 score, the proposed formulation outperforms other methods on 10 data sets. The low performance of ASO$_{\text{SVM}}$ may be due to the early termination of its iterative procedure in parameter tuning, since it is computationally very expensive. In general, tuning the parameter C in SVM with cross-validation yield a performance improvement of about 1% in most cases. The performance improvement achieved by the proposed formulation over other compared methods is consistent across data sets and performance measures.

[1] `http://www.public.asu.edu/~sji03/multilabel/`

Table 3: Summary of performance for the 6 compared methods on the first 6 Yahoo data sets in terms of AUC (top section), macro F1 (middle section), and micro F1 (bottom section). All parameters of the 6 methods are tuned by cross-validation, and the averaged performance over 5 random sampling of training instances is reported. The highest performance is highlighted in each case.

Algorithm	Arts	Business	Computer	Education	Entertainment	Health
ML_{LS}	**0.7711±0.0040**	**0.8348±0.0050**	**0.7964±0.0050**	**0.7753±0.0071**	**0.8264±0.0018**	0.8483±0.0043
CCA+Ridge	0.7571±0.0045	0.8216±0.0066	0.7932±0.0071	0.7597±0.0099	0.8043±0.0083	0.8477±0.0152
CCA+SVM	0.7519±0.0037	0.8169±0.0072	0.7774±0.0121	0.7581±0.0096	0.7965±0.0101	0.8409±0.0087
ASO_{SVM}	0.7678±0.0037	0.8261±0.0067	0.7847±0.0064	0.7446±0.0095	0.8207±0.0033	0.8621±0.0042
SVM_{C}	0.7674±0.0046	0.8263±0.0058	0.7943±0.0054	0.7685±0.0062	0.8155±0.0040	0.8617±0.0039
SVM	0.7668±0.0045	0.8271±0.0055	0.7940±0.0061	0.7716±0.0067	0.8177±0.0035	**0.8634±0.0040**
ML_{LS}	**0.3583±0.0076**	**0.3985±0.0052**	**0.3219±0.0241**	**0.3864±0.0096**	**0.4874±0.0121**	**0.5966±0.0115**
CCA+Ridge	0.3190±0.0090	0.3779±0.0114	0.2799±0.0366	0.3602±0.0142	0.4352±0.0160	0.5431±0.0363
CCA+SVM	0.3158±0.0093	0.3758±0.0127	0.3059±0.0304	0.3618±0.0125	0.4409±0.0172	0.5338±0.0343
ASO_{SVM}	0.3568±0.0094	0.3736±0.0120	0.2873±0.0265	0.3262±0.0127	0.4344±0.0154	0.5814±0.0059
SVM_{C}	0.3216±0.0090	0.3533±0.0103	0.2609±0.0292	0.3588±0.0067	0.4260±0.0047	0.5632±0.0059
SVM	0.3382±0.0130	0.3677±0.0070	0.2948±0.0282	0.3830±0.0146	0.4462±0.0103	0.5705±0.0078
ML_{LS}	**0.4716±0.0070**	**0.7645±0.0056**	**0.5585±0.0097**	**0.4899±0.0097**	**0.5901±0.0098**	**0.6809±0.0078**
CCA+Ridge	0.4444±0.0186	0.7508±0.0068	0.5414±0.0051	0.4538±0.0127	0.5506±0.0260	0.6771±0.0053
CCA+SVM	0.4524±0.0075	0.7528±0.0074	0.5394±0.0198	0.4647±0.0166	0.5498±0.0249	0.6804±0.0098
ASO_{SVM}	0.4449±0.0078	0.7384±0.0075	0.4305±0.0219	0.4322±0.0124	0.5605±0.0113	0.6754±0.0076
SVM_{C}	0.4449±0.0024	0.7384±0.0091	0.5275±0.0112	0.4745±0.0045	0.5413±0.0092	0.6714±0.0105
SVM	0.4574±0.0057	0.7584±0.0073	0.5458±0.0131	0.4773±0.0092	0.5701±0.0108	0.6773±0.0051



Health (with a fixed parameter) Health (with parameter tuning) Science (with a fixed parameter) Science (with parameter tuning)

Figure 1: Comparison of computation time for ML_{LS}, SVM, and ASO_{SVM} on the Health (left two panels) and Science (right two panels) data sets. The computation time for a fixed parameter setting and that for parameter tuning using cross-validation are both depicted for each data set. See the text for more details.

5.3 Scalability Evaluation

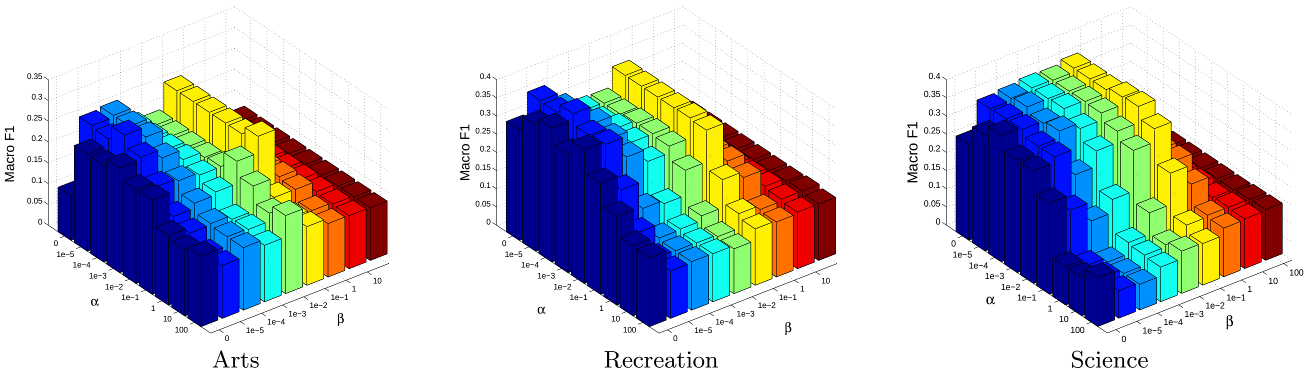
We evaluate the scalability of the proposed multi-label formulation on the Health and Science data sets which contain the minimum and maximum number of labels among the 11 data sets. In particular, we increase the number of training samples on the Health and Science data sets gradually, and record the computation time of ML_{LS}, SVM, and ASO_{SVM}. The training time for a fixed parameter setting and the time for parameter tuning using cross-validation are plotted in Figure 1. We can observe that SVM is the fastest and ASO_{SVM} is the slowest among the three compared algorithms. Moreover, the difference between ML_{LS} and SVM is small. The computational cost of the proposed formulation is dominated by the cost of SVD computation on the data matrix X, and it is independent of the number of labels. In contrast, the computational costs of SVM and ASO_{SVM} depend on the number of labels. Hence, the difference between SVM and ML_{LS} tends to be smaller on the Science data set, since the Science data set has a larger number of labels than the Health data set (14 and 22 labels, respectively). Note that in ML_{LS}, the two regularization parameters α and β are tuned using double cross-validation. However, the SVD on X needs to be computed only once irrespective of the size of the candidate sets for α and β. This experiment also shows that the running time of ASO_{SVM} may fluctuate as the number of training instances increases. This may be due to the fact that the convergence rate of the ASO_{SVM} algorithm depends on the initialization.

5.4 Sensitivity Study

We conduct experiments to evaluate the sensitivity of the proposed formulation to the values of the regularization parameters α and β. We randomly sample 1000 data points from each of the three data sets Arts, Recreation, and Science, and the averaged macro F1 scores over 5-fold cross-validation for different values of α and β are depicted in Figure 2. We can observe that the highest performance on all three data sets is achieved at some intermediate values of α and β. Moreover, this experiments show that the performance of the proposed multi-label formulation is sensitive to the values of the regularization parameters. Note that the parameter tuning time of the proposed formulation does not depend on the size of the candidate sets directly, since

Table 4: Summary of performance for the 6 compared methods on the last 5 Yahoo data sets in terms of AUC (top section), macro F1 (middle section), and micro F1 (bottom section). All parameters of the 6 methods are tuned by cross-validation, and the averaged performance over 5 random sampling of training instances is reported. The highest performance is highlighted in each case.

Algorithm	Recreation	Reference	Science	Social	Society
$\mathrm{ML}_{\mathrm{LS}}$	**0.8202±0.0064**	**0.8358±0.0026**	**0.8332±0.0037**	0.8320±0.0038	**0.7347±0.0030**
CCA+Ridge	0.8106±0.0072	0.8260±0.0109	0.8161±0.0020	0.8189±0.0032	0.7204±0.0060
CCA+SVM	0.7896±0.0119	0.8054±0.0056	0.7946±0.0052	0.7448±0.0077	0.7007±0.0088
$\mathrm{ASO}_{\mathrm{SVM}}$	0.8123±0.0054	0.8340±0.0041	0.8073±0.0042	0.7942±0.0092	0.7321±0.0021
$\mathrm{SVM}_{\mathrm{C}}$	0.8092±0.0063	0.8345±0.0025	0.8277±0.0015	**0.8392±0.0026**	0.7308±0.0030
SVM	0.8157±0.0068	0.8327±0.0052	0.8311±0.0012	0.8377±0.0033	0.7340±0.0024
$\mathrm{ML}_{\mathrm{LS}}$	**0.4519±0.0138**	**0.4208±0.0078**	**0.4337±0.0042**	**0.3680±0.0223**	**0.3369±0.0164**
CCA+Ridge	0.4286±0.0182	0.3330±0.0120	0.3577±0.0074	0.3279±0.0059	0.2961±0.0274
CCA+SVM	0.4331±0.0158	0.3417±0.0139	0.3650±0.0093	0.3126±0.0134	0.2996±0.0146
$\mathrm{ASO}_{\mathrm{SVM}}$	0.4136±0.0133	0.4116±0.0102	0.3397±0.0109	0.3017±0.0017	0.3023±0.0132
$\mathrm{SVM}_{\mathrm{C}}$	0.3852±0.0233	0.3795±0.0119	0.3770±0.0181	0.3232±0.0160	0.2919±0.0191
SVM	0.4460±0.0084	0.4005±0.0084	0.4093±0.0174	0.3380±0.0190	0.3069±0.0142
$\mathrm{ML}_{\mathrm{LS}}$	**0.5351±0.0023**	**0.6020±0.0067**	**0.5254±0.0040**	0.6606±0.0059	**0.4874±0.0142**
CCA+Ridge	0.5223±0.0069	0.5336±0.0377	0.4704±0.0122	**0.6607±0.0045**	0.4783±0.0273
CCA+SVM	0.5159±0.0066	0.5448±0.0315	0.4815±0.0036	0.6012±0.0183	0.4690±0.0123
$\mathrm{ASO}_{\mathrm{SVM}}$	0.4976±0.0078	0.5580±0.0047	0.4564±0.0128	0.6492±0.0112	0.4639±0.0025
$\mathrm{SVM}_{\mathrm{C}}$	0.4797±0.0145	0.5856±0.0053	0.4774±0.0169	0.6500±0.0095	0.4569±0.0108
SVM	0.5284±0.0049	0.6002±0.0064	0.5142±0.0085	0.6573±0.0123	0.4801±0.0127



Arts Recreation Science

Figure 2: The change of macro F1 scores as the regularization parameters α and β vary in the range $[0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100]$ for the Arts (left panel), Recreation (middle panel), and Science (right panel) data sets.

the computational cost is dominated by that of the SVD of X which needs to be performed only once. Hence, large candidate sets for α and β can be employed in practice.

6. CONCLUSION AND DISCUSSION

We present a framework for extracting shared subspaces in multi-label classification in this paper. In this framework, a subspace is assumed to be shared among multiple labels, and a linear transformation is computed to discover this subspace. We show that when the least squares loss is used in classification, the optimal solution for the proposed formulation can be computed via a generalized eigenvalue problem. For high-dimensional data, direct computation of this problem is computationally expensive, and we develop an efficient algorithm for this case. We show that the proposed formulation is a general framework that includes several well-known formulations as special cases. Experimental results on 11 multi-topic web page categorization tasks show that the proposed formulation outperforms competing methods in most cases.

Out results show that applying regularization on both parts of the predictor can potentially improve performance. We have attempted to compare the proposed formulation with an extension of the ASO algorithm in which both parts of the predictor are regularized. However, this extension of the ASO algorithm is computationally demanding when both regularization parameters are tuned using double cross-validation. Hence, we are not able to include its results in this paper. We will explore ways to improve the efficiency of this algorithm in the future. The data matrices in many applications such as the ones used in this paper are sparse. Hence, techniques for computing the SVD of sparse matrices as proposed in [20] can be employed to expedite the computation. We plan to apply such techniques in our algorithm in the future.

7. REFERENCES

[1] Y. Amit, M. Fink, N. Srebro, and S. Ullman. Uncovering shared structures in multiclass classification. In *Proceedings of the 24th International Conference on Machine Learning*, pages 17–24, 2007.

[2] E. D. Andersen and K. D. Andersen. The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In *High Performance Optimization*, pages 197–232. Kluwer Academic Publishers, 2000.

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

[4] J. Arenas-García, K. B. Petersen, and L. K. Hansen. Sparse kernel orthonormalized PLS for feature extraction in large data sets. In *Advances in Neural Information Processing Systems 19*, pages 33–40. 2007.

[5] B. Bakker and T. Heskes. Task clustering and gating for Bayesian multitask learning. *Journal of Machine Learning Research*, 4:83–99, 2003.

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

[7] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001.

[8] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems 14*, pages 681–687, 2002.

[9] R.-E. Fan and C.-J. Lin. A study on threshold selection for multi-label classification, 2007.

[10] K. Fukunaga. *Introduction to statistical pattern recognition*. Academic Press Professional, 2nd edition, 1990.

[11] G. M. Fung and O. L. Mangasarian. Multicategory proximal support vector machine classifiers. *Machine Learning*, 59(1-2):77–97, 2005.

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

[13] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, 1996.

[14] A. Hoerl and R. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(3):55–67, 1970.

[15] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3-4):321–377, 1936.

[16] R. Jin and Z. Ghahramani. Learning with multiple labels. In *Advances in Neural Information Processing Systems 15*, pages 897–904. 2002.

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

[18] F. Kang, R. Jin, and R. Sukthankar. Correlated label propagation with application to multi-label learning. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1719–1726, 2006.

[19] H. Kazawa, T. Izumitani, H. Taira, and E. Maeda. Maximal margin labeling for multi-topic text categorization. In *Advances in Neural Information Processing Systems 17*, pages 649–656. 2005.

[20] R. M. Larsen. Computing the SVD for large and sparse matrices, 2000.

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

[22] A. McCallum. Multi-label text classification with a mixture model trained by EM. In *AAAI Workshop on Text Learning*, 1999.

[23] H. Park, M. Jeon, and J. B. Rosen. Lower dimensional representation of text data based on centroids and least squares. *BIT*, 43(2):1–22, 2003.

[24] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

[25] V. Roth and B. Fischer. Improved functional prediction of proteins by learning kernel combinations in multilabel settings. *BMC Bioinformatics*, 8:S12, 2007.

[26] S. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines,Regularization, Optimization and Beyond*. MIT Press, 2002.

[27] N. Ueda and K. Saito. Parametric mixture models for multi-labeled text. In *Advances in Neural Information Processing Systems 15*, pages 721–728. 2002.

[28] N. Ueda and K. Saito. Single-shot detection of multiple categories of text using parametric mixture models. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 626–631, 2002.

[29] H. Wold. Estimation of principal components and related models by iterative least squares. In *P.R. Krishnaiah, editor, Multivariate Analysis*, pages 391–420. Academic Press, New York, 1966.

[30] R. Yan, J. Tesic, and J. R. Smith. Model-shared subspace boosting for multi-label classification. In *Proceedings of the thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 834–843, 2007.

[31] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 412–420, 1997.

[32] J. Ye. Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. *Journal of Machine Learning Research*, 6:483–502, 2005.

[33] K. Yu, S. Yu, and V. Tresp. Multi-label informed latent semantic indexing. In *Proceedings of the twenty-eighth Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 258–265, 2005.

[34] J. Zhang, Z. Ghahramani, and Y. Yang. Learning multiple related tasks using latent independent component analysis. In *Advances in Neural Information Processing Systems 18*.