Correlated Multi-Label Feature Selection

Quanquan Gu
Dept. of Computer Science
University of Illinois at Urbana-Champaign
IL, 61801, USA
qgu3@illinois.edu

Zhenhui Li
Dept. of Computer Science
University of Illinois at Urbana-Champaign
IL, 61801, USA
zli28@uiuc.edu

Jiawei Han
Dept. of Computer Science
University of Illinois at Urbana-Champaign
IL, 61801, USA
hanj@cs.uiuc.edu

ABSTRACT

Multi-label learning studies the problem where each instance is associated with a set of labels. There are two challenges in multi-label learning: (1) the labels are interdependent and correlated, and (2) the data are of high dimensionality. In this paper, we aim to tackle these challenges in one shot. In particular, we propose to learn the label correlation and do feature selection simultaneously. We introduce a matrix-variate Normal prior distribution on the weight vectors of the classifier to model the label correlation. Our goal is to find a subset of features, based on which the label correlation regularized loss of label ranking is minimized. The resulting multi-label feature selection problem is a mixed integer programming, which is reformulated as quadratically constrained linear programming (QCLP). It can be solved by cutting plane algorithm, in each iteration of which a minimax optimization problem is solved by dual coordinate descent and projected sub-gradient descent alternatively. Experiments on benchmark data sets illustrate that the proposed methods outperform single-label feature selection method and many other state-of-the-art multi-label learning methods.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.5.1 [**Pattern Recognition**]: Models

General Terms

Algorithms, Experimentation

Keywords

Feature Selection, Label Correlation, Multi-Label Learning, Dual Coordinate Descent, Cutting Plane

1. INTRODUCTION

Multi-label learning [7, 30, 35, 39, 13, 27, 4, 11, 31, 36] is a very important topic in data mining and information retrieval. It studies the problem where each instance is associated with a set of labels. This is not uncommon in many important applications, such as protein function classification [7], text categorization [19], and semantic scene classification [1]. For example, one gene can be associated with several functions, one image may have several tags, and one document can cover several topics.

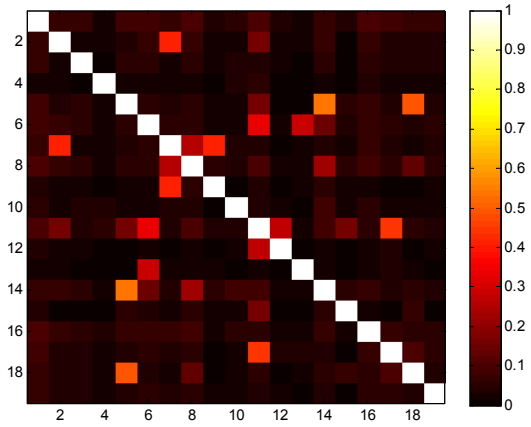

Figure 1: The label correlation computed from the labels of all the data points in the Yahoo/Arts data set

There are mainly two challenges in multi-label learning. First, different from traditional single-label learning where the classes are mutually exclusive, the classes in multi-label learning are typically interdependent and correlated, which poses more difficulties to predict all the relevant labels for a given instance. For example, in image annotation, "sea" and "ship" tend to appear in the same image, while "car" typically does not appear together with "ship". Figure 1 illustrates the label correlation which is computed from the labels of all the data points in the Yahoo/Arts data set. The higher the value between two labels is (lighter color), the more correlated these two labels are. It can be seen that the 5th label is highly correlated with the 14th and 18th labels, while it is not correlated with the 12th label. On the other hand, the label correlation offers a possibility to infer the unknown label of an instance from the known label. In order to utilize the relation between labels, [7, 6] proposed to learn the ranks of labels for each instance, which is basically first-order information. However, the correlation among labels is second-order information [36]. And we will show that it is essential for better performance in our experiments.

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

The second challenge is that multi-labeled data usually have thousands or even tens of thousands of features. This is especially true for documents and news articles. For example, the news articles in the Yahoo data set used in our experiments are of about 20K features. As we know, high dimensional data may cause the *curse of dimensionality*, which increases the computational burden and deteriorate the generalization ability of the classifier. To overcome this problem, many dimensionality reduction based multi-label learning approaches [35, 39, 13, 31] have been proposed. Although these methods perform good for high dimensional data, they still fail to explicitly model the label correlation, which is crucial for better performance.

In this paper, based on the above motivation, we aim to solve the two challenges in one shot. We built up our model on the label rank support vector machine (LaRank SVM) [7][1], which is among state-of-the-art multi-label learning methods [7, 13]. We introduce a matrix-variate Normal prior distribution [10] on the weight vectors of LaRank SVM. Since the column covariance matrix of matrix-variate Normal distribution characterizes the correlation among the weight vectors, each of which associates with one label, the label correlation is modeled explicitly. To avoid the curse of dimensionality, we incorporate feature selection into LaRank SVM. Our goal is to find a subset of features, based on which the label correlation regularized loss of label ranking [7] is minimized. The resulting multi-label feature selection is a mixed integer programming problem, which is difficult to solve. Fortunately, it can be reformulated as Quadratically Constrained Linear Programming (QCLP) [2]. It is solved by cutting plane algorithm [17], in each iteration of which a minimax optimization problem is solved by dual coordinate descent [12] and projected sub-gradient descent [22] alternatively. As a by-product, we also propose a correlated label rank support vector machine (CLaRank SVM), which is an extension of LaRank SVM [7]. It is worth noting that, the proposed approach is able to not only learn the label correlation automatically, but also reduce the dimensionality of the original data. Experiments on benchmark data sets indicate that the proposed methods outperform single-label feature selection and many other state-of-the-art methods.

The remainder of this paper is organized as follows. In Section 2, we discuss several related works on multi-label learning. In Section 3 we present correlated multi-label feature selection. The experiments on real world data sets are demonstrated in Section 4. Finally, we draw a conclusion and point out the future work in Section 5.

1.1 Notation

In multi-label learning with c labels, each data point $\mathbf{x}_i \in \mathbb{R}^d, i = 1, \ldots, n$ can be associated with a set of labels, i.e., $y_i \subseteq \{1, 2, \ldots, c\}$. We denote the complementary set of y_i by $\bar{y}_i$, and the cardinality of y_i by $|y_i|$. For example, suppose there are totally 6 labels, and $\mathbf{x}_i$ is labeled by the 2nd and 3rd labels, then $y_i = \{2, 3\}$, $\bar{y}_i = \{1, 4, 5, 6\}$ and $|y_i| = 2$. $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ represents the data matrix. $\mathbf{x}^j$ denotes the jth row of $\mathbf{X}$. $\mathbf{e}_i$ is a unit vector of all zeros except the ith element equal to 1. $\mathbf{1}$ is a vector of all ones. $\mathbf{0}$ is a vector of all zeros. Given a matrix $\mathbf{R}$, we denote its (k, l)th entry by R_{kl}, and its inverse matrix by $\mathbf{R}^{-1}$. R^{-1}_{kl} is the (k, l)th entry of $\mathbf{R}^{-1}$.

[1]Note that in the original paper, the authors called this model as Rank SVM. To distinguish this model from another well-known Rank SVM proposed in [14] for information retrieval, we call this model as Label Rank SVM or simply LaRank SVM because it ranks the labels rather than data points.

2. RELATED WORK

In this section, we give a brief review of multi-label classification methods which are related to ours. Existing multi-label learning methods can be cast into different families.

The first family of multi-label learning method is to divide multi-label learning into a set of one-against-all binary classification problems [23]. However, since each label is treated independently, it fails to consider the correlation among different labels, which is essential in multi-label learning. It is desirable for a multi-label learning method to make use of label correlation for better performance. Moreover, this approach suffers from imbalanced data when constructing binary classifiers to distinguish each class from the remaining classes. This problem becomes more severe when the number of classes is large.

The second category of multi-label learning approaches are based on *Label Ranking* [7, 6, 4], where ranking-based strategy is taken to learn a ranking function of labels from the labeled instances and apply it to obtain a real-valued score for each instance-label pair, then classify each instance by choosing all the labels whose scores are above the given threshold. They achieve state-of-the-art results and are scalable to large-scale data with the recent progress in support vector machine optimization [15, 25, 12, 16]. However, these methods do not explicitly exploit the label correlation, and are suffering from curse of dimensionality. This motivates us to propose a model built up on LaRank SVM [7], while it is able to overcome the limitations.

Another family of multi-label learning methods are based on dimensionality reduction, which assume that all the labels share a common subspace. For example, [35] extended unsupervised latent sematic indexing to make use of multi-label information. [39] proposed Multi-label Dimensionality reduction via Dependence Maximization (MDDM) method to identify a lower-dimensional subspace by maximizing the dependence between the original features and associated class labels. [31] proposed Multi-Label Linear Discriminant Analysis (MLDA) which is an extension of linear discriminant analysis. [27] proposed to construct a hyper-graph on both the data points and labels, and find a subspace to preserve the information of the hyper-graph for classification. [13] proposed Multi-Label Least Square (MLLS) method to extract a common subspace shared among multiple labels. These methods have been proved very effective for multi-label learning. They are also able to deal with the curse of dimensionality. Subspace based methods utilize the correlation between data and labels. However, they do not consider the correlation among labels, which is not uncommon in multi-label data as shown before. In this paper, rather than subspace learning, we propose to do feature selection for multi-label learning, which can be integrated into LaRank SVM [7] coherently. We assume that all the labels share a common subset of features. Due to the close relationship between subspace learning and feature selection, feature selection plays a similar role as subspace learning. Moreover, to capture the correlation among labels, we propose to learn the label correlation at the same time as feature selection.

3. THE PROPOSED METHOD

Since the proposed method is built upon LaRank SVM [7], we first briefly review the formulation of LaRank SVM. It borrowed the large margin idea to multi-label learning and modified SVM to a ranking system of labels. The basic idea is, for each instance, the ranking scores of the labels assigned to the instance should be higher than the ranking scores of the labels not assigned to it. The resulting maximum margin multi-label ranking system is

$$\begin{aligned} \min_{\mathbf{w},\boldsymbol{\xi}} \quad & \frac{1}{2}\sum_{k=1}^{c}||\mathbf{w}_k||^2 + C\sum_{i=1}^{n}\frac{1}{|y_i||\bar{y}_i|}\sum_{(k,l)\in y_i\times\bar{y}_i}\xi_{ikl} \\ s.t. \quad & \langle \mathbf{w}_k^T - \mathbf{w}_l^T, \mathbf{x}_i\rangle \geq 1-\xi_{ikl}, (k,l)\in y_i\times\bar{y}_i \\ & \xi_{ikl}\geq 0, i=1,\ldots,n, \end{aligned} \tag{1}$$

where $C>0$ is a regularization parameter. Note that a bias term can be incorporated into the form by expanding the weight vector and input feature vector as $\mathbf{w}_k \leftarrow [\mathbf{w}_k^T, b_k]^T$ and $\mathbf{x} \leftarrow [\mathbf{x}^T, 1]^T$. It has been shown that LaRank SVM performs better than SVM [7, 27]. LaRank SVM considers the ordinal relation among labels, which is first-order information. However, LaRank SVM does not consider the label correlation, which is a kind of second-order information among labels. Moreover, when the dimensionality of the data is very high, LaRank SVM does not perform as good as some dimensionality reduction based methods [27, 13]. This motivates us to consider the label correlation and do feature selection (dimensionality reduction) simultaneously in this paper.

3.1 Incorporating Label Correlation

To capture the correlation between labels, we place a matrix-variate Normal distribution prior [10] on the weight vectors of LaRank SVM, i.e., $\mathbf{W}=[\mathbf{w}_1,\ldots,\mathbf{w}_c]$,

$$p(\mathbf{W}|\mathbf{D}) = \mathcal{MN}(\mathbf{0}_{d\times c}, \mathbf{I}_d\otimes\mathbf{D}), \tag{2}$$

where $\mathbf{0}_{d\times c}$ denotes a $d\times c$ zero matrix, $\mathbf{I}_d$ is a $d\times d$ identity matrix, and $\mathcal{MN}(\mathbf{X}|\mathbf{M},\mathbf{A}\otimes\mathbf{B})$ denotes a matrix-variate normal distribution with mean $\mathbf{M}\in\mathbb{R}^{a\times b}$, row covariance matrix $\mathbf{A}\in\mathbb{R}^{a\times a}$, and column covariance matrix $\mathbf{B}\in\mathbb{R}^{b\times b}$. The probability density function of the matrix-variate normal distribution is defined as

$$\begin{aligned} & p(\mathbf{X}|\mathbf{M},\mathbf{A},\mathbf{B}) \\ = \quad & \frac{\exp\left\{-\frac{1}{2}\mathrm{tr}(\mathbf{A}^{-1}(\mathbf{X}-\mathbf{M})\mathbf{B}^{-1}(\mathbf{X}-\mathbf{M})^T)\right\}}{(2\pi)^{ab/2}|\mathbf{A}|^{b/2}|\mathbf{B}|^{a/2}}. \end{aligned} \tag{3}$$

Note that similar prior has been used for multi-task learning [37] and transfer distance metric learning [38]. Plug Eq.(3) into Eq.(2), it can be simplified as

$$p(\mathbf{W}|\mathbf{D}) = \frac{\exp\left\{-\frac{1}{2}\mathrm{tr}(\mathbf{W}\mathbf{D}^{-1}\mathbf{W}^T)\right\}}{(2\pi)^{dc/2}|\mathbf{D}|^{d/2}}. \tag{4}$$

Since the column covariance matrix $\mathbf{D}$ models the correlation between any two $\mathbf{w}_k$, it is able to capture the correlation of different labels.

3.2 Multi-Label Feature Selection

As to feature selection, we introduce a binary variable $p_j\in\{0,1\}, j=1,\ldots,d$ for each feature, such that if $p_j=1$, then the jth feature is selected. Otherwise, it is discarded. Our goal is to find a subset of features, such that the the label correlation regularized loss of the LaRank SVM in Eq.(1) is minimized,

$$\begin{aligned} \min_{\mathbf{w},\mathbf{D},\boldsymbol{\xi},\mathbf{p}} \quad & \frac{1}{2}\sum_{k=1}^{c}||\mathbf{w}_k||_2^2 + C\sum_{i=1}^{n}\frac{1}{|y_i||\bar{y}_i|}\sum_{(k,l)\in y_i\times\bar{y}_i}\xi_{ikl} \\ & -\frac{\mu}{2}\log p(\mathbf{W}|\mathbf{D}), \\ s.t. \quad & \langle \mathbf{w}_k^T - \mathbf{w}_l^T, \mathbf{p}\circ\mathbf{x}_i\rangle \geq 1-\xi_{ikl}, (k,l)\in y_i\times\bar{y}_i \\ & \xi_{ikl}\geq 0, i=1,\ldots,n, \\ & \mathbf{p}\in\{0,1\}^d, \mathbf{p}^T\mathbf{1}=m, \end{aligned} \tag{5}$$

where $\mathbf{p}\circ\mathbf{x}_i$ is element-wise Hadamard product which performs feature selection. Note that the first two terms in the objective function can be seen as the negative log likelihood of some kind of distribution on $\mathbf{w}_k$, and the third term is the negative logarithm of a prior on $\mathbf{w}_k$. Hence the whole objective function can be seen as a negative logarithm of the posterior on $\mathbf{w}_k$. This is in spirit the same as maximum a posterior principle. Unfortunately, the term $-\frac{\mu}{2}\log p(\mathbf{W}|\mathbf{D}) \propto \frac{1}{2}\mathrm{tr}(\mathbf{W}\mathbf{D}^{-1}\mathbf{W}^T) - \frac{d}{2}\log|\mathbf{D}| - \frac{dc}{2}\log(2\pi)$ is not easy to optimize since it is non-convex. In this paper, we turn to solve the following similar problem,

$$\begin{aligned} \min_{\mathbf{w},\mathbf{D},\boldsymbol{\xi},\mathbf{p}} \quad & \frac{1}{2}\sum_{k=1}^{c}||\mathbf{w}_k||_2^2 + C\sum_{i=1}^{n}\frac{1}{|y_i||\bar{y}_i|}\sum_{(k,l)\in y_i\times\bar{y}_i}\xi_{ikl} \\ & +\frac{\mu}{2}\mathrm{tr}(\mathbf{W}\mathbf{D}^{-1}\mathbf{W}^T), \\ s.t. \quad & \langle \mathbf{w}_k^T - \mathbf{w}_l^T, \mathbf{p}\circ\mathbf{x}_i\rangle \geq 1-\xi_{ikl}, (k,l)\in y_i\times\bar{y}_i \\ & \xi_{ikl}\geq 0, i=1,\ldots,n, \\ & \mathbf{p}\in\{0,1\}^d, \mathbf{p}^T\mathbf{1}=m, \\ & \mathbf{D}\succeq 0, \mathrm{tr}(\mathbf{D})=1, \end{aligned} \tag{6}$$

We call Eq.(6) as *Correlated Multi-Label Feature Selection* (CMLFS).

Both Eq. (5) and Eq.(6) are mixed integer programming [2]. Compared with Eq.(5), although Eq.(6) sacrifices the sound probabilistic interpretation to some extent, it has a more desirable optimization property, which is stated in the following theorems.

Theorem 3.1 *Given* $\mathbf{p}$*, the optimization problem in Eq.(6) is jointly convex in* $\mathbf{w}_k$ *and* $\mathbf{D}$*.*

PROOF. Please refer to [37]. □

It is worth noting that there are two special instances of the proposed model in Eq.(6). First, if we set $\mu=0$, then the model in Eq.(6) reduces to multi-label feature selection without learning the label correlation. In the rest of this paper, we refer to this special model as *Multi-Label Feature Selection* (MLFS). Second, if we fix $\mathbf{p}=\mathbf{1}$, then the model in Eq.(6) can be seen as an extension of label rank SVM, which is not only learning to rank the labels, but also learning the correlation among labels. This special model is referred to as *Correlated Label Rank SVM* (CLaRank SVM). It is obvious that if we set $\mu=0$ and fix $\mathbf{p}=\mathbf{1}$ simultaneously, then Eq.(6) exactly reduces to original label rank SVM in Eq.(1). In the sequel, we will present the optimization algorithm to solve Eq. (6).

3.3 The Dual Problem

Instead of directly optimizing the problem in Eq. (6), we choose to optimize its dual problem [2].

Theorem 3.2 *The dual of the problem in Eq. (6) is*

$$\begin{aligned}\min_{\mathbf{D},\mathbf{p}}\max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{n}\sum_{(k,l)\in y_i\times\bar{y}_i}\alpha_{ikl} \\ & -\frac{1}{2}\sum_{k,l=1}^{c}R_{kl}^{-1}\sum_{i,j=1}^{n}\beta_{ik}\beta_{jl}(\mathbf{p}\circ\mathbf{x}_i)^T(\mathbf{p}\circ\mathbf{x}_j), \\ s.t. \quad & 0\le\alpha_{ikl}\le\frac{C}{|y_i||\bar{y}_i|} \\ & \mathbf{p}\in\{0,1\}^d, \mathbf{p}^T\mathbf{1}=m \\ & \mathbf{D}\succeq 0, tr(\mathbf{D})=1, \end{aligned} \tag{7}$$

where $\mathbf{R}=\mathbf{I}+\mu\mathbf{D}^{-1}$, $\mathbf{R}^{-1}$ *is the inverse matrix of* $\mathbf{R}$, *and* R_{kl}^{-1} *is the* (k,l)*th entry of* $\mathbf{R}^{-1}$, β_{ik} *is defined as*

$$\beta_{ik}=\sum_{(p,q)\in y_i\times\bar{y}_i}\gamma_{ipq}^{k}\alpha_{ipq}, \tag{8}$$

and γ_{ipq}^{k} *is defined as,*

$$\gamma_{ipq}^{k}=\begin{cases}1, & if\ p=k\\ -1, & if\ q=k\\ 0, & if\ p\neq k\ and\ q\neq k\end{cases}. \tag{9}$$

Moreover, we have

$$\sum_{l=1}^{c}R_{kl}\mathbf{w}_l=\sum_{i=1}^{n}\beta_{ik}\mathbf{p}\circ\mathbf{x}_i. \tag{10}$$

For the sake of notional simplicity, we denote the objective function of Eq.(7) by $f(\boldsymbol{\alpha},\mathbf{D},\mathbf{p})$, and define

$$\begin{aligned}\mathcal{D} &= \{\mathbf{D}\succeq 0,\mathrm{tr}(\mathbf{D})=1\},\\ \mathcal{P} &= \left\{\mathbf{p}\in\{0,1\}^d,\mathbf{p}^T\mathbf{1}=m\right\},\\ \mathcal{A} &= \left\{\boldsymbol{\alpha}|0\le\alpha_{ikl}\le\frac{C}{|y_i||\bar{y}_i|},k=1,\ldots,c\right\}.\end{aligned} \tag{11}$$

Then the optimization problem in Eq.(7) can be rewritten as

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{p}\in\mathcal{P}}\max_{\boldsymbol{\alpha}\in\mathcal{A}}f(\boldsymbol{\alpha},\mathbf{D},\mathbf{p}). \tag{12}$$

By interchanging the order of $\min_{\mathbf{D}\in\mathcal{D},\mathbf{p}\in\mathcal{P}}$ and $\max_{\boldsymbol{\alpha}_k\in\mathcal{A}_k}$ in Eq. (12), we obtain

$$\max_{\boldsymbol{\alpha}\in\mathcal{A}}\min_{\mathbf{D}\in\mathcal{D},\mathbf{p}\in\mathcal{P}}f(\boldsymbol{\alpha},\mathbf{D},\mathbf{p}). \tag{13}$$

According to the minimax theorem [18], the optimal objective value of Eq.(7) is an upper bound of that of Eq.(13).

The problem in Eq. (13) is indeed a convex-concave optimization problem, and therefore its optimal solution is a saddle point for the function $f(\boldsymbol{\alpha},\mathbf{D},\mathbf{p})$ subject to the constraints in Eq. (11). Let $(\boldsymbol{\alpha}^*,\mathbf{D}^*,\mathbf{p}^*)$ be optimal to Eq. (13). For any feasible $\boldsymbol{\alpha}$ and $\mathbf{p}$, we have

$$f(\boldsymbol{\alpha},\mathbf{D}^*,\mathbf{p}^*)\le f(\boldsymbol{\alpha}^*,\mathbf{D}^*,\mathbf{p}^*)\le f(\boldsymbol{\alpha}^*,\mathbf{D},\mathbf{p}). \tag{14}$$

Borrowing the idea used in [5] [20] [28], we add an additional variable $\theta\in\mathbb{R}$, then the problem in Eq. (13) can be reformulated equivalently as follows

$$\begin{aligned}&\max_{\boldsymbol{\alpha}\in\mathcal{A}}\min_{\mathbf{D}\in\mathcal{D}}\max_{\theta}-\theta\\ &\text{s.t.}\theta\ge -f(\boldsymbol{\alpha},\mathbf{D},\mathbf{p}^t),\mathbf{p}^t\in\mathcal{P}.\end{aligned} \tag{15}$$

Note that each $\mathbf{p}^t\in\mathcal{P}$ corresponds to one constraint, so the above optimization problem has $\binom{d}{m}$ constraints. The optimization problem in Eq.(15) is called Quadratically Constrained Linear Programming (QCLP) [2].

We introduce a set of Lagrange multipliers $\lambda_t\ge 0$, each of which corresponds to an inequality constraint $\theta\ge -f(\boldsymbol{\alpha},\mathbf{D},\mathbf{p}^t)$. Then the Lagrange function of Eq.(6) is given by

$$\mathcal{L}(\theta)=-\theta+\sum_{t=1}^{|\mathcal{P}|}\lambda_t(\theta+f(\boldsymbol{\alpha},\mathbf{D},\mathbf{p}^t)). \tag{16}$$

Taking the partial derivative of $\mathcal{L}$ with respect θ and setting it to zero, we obtain

$$\frac{\partial\mathcal{L}}{\partial\theta}=-1+\sum_{t=1}^{|\mathcal{P}|}\lambda_t=0. \tag{17}$$

Plugging Eq.(17) back into Eq.(15), we get the dual problem of the inner maximization problem in Eq(15), we obtain the following problem,

$$\begin{aligned}&\max_{\boldsymbol{\alpha}\in\mathcal{A}}\min_{\mathbf{D}\in\mathcal{D},\lambda_t\in\Lambda}\sum_{t=1}^{|\mathcal{P}|}\lambda_t f(\boldsymbol{\alpha},\mathbf{D},\mathbf{p}^t)\\ =\ &\min_{\mathbf{D}\in\mathcal{D},\lambda_t\in\Lambda}\max_{\boldsymbol{\alpha}\in\mathcal{A}}\sum_{t=1}^{|\mathcal{P}|}\lambda_t f(\boldsymbol{\alpha},\mathbf{D},\mathbf{p}^t),\end{aligned} \tag{18}$$

where $\Lambda=\{\lambda_t|\sum_{t=1}^{|\mathcal{P}|}\lambda_t=1,\lambda_t\ge 0\}$. The equality holds due to the fact that the objective function is concave in $\boldsymbol{\alpha}$ and convex in $\mathbf{D}$ and $\boldsymbol{\lambda}$.

3.4 Alternating Optimization

Actually, Eq. (18) can be seen as a multiple kernel learning problem [22], where the base kernels and kernel weights are the same for all the labels. It is worth noting that [29] proposed a multiple kernel learning with multiple labels. Their method is different from ours. In their method, the kernel weights are different for each label. Moreover, they did not consider the label correlation. As a result, their algorithm cannot be adapted to our problem.

Following the technique used in the state-of-the-art single-label multiple kernel learning [22], we optimize Eq. (18) in an alternative way. In particular, we alternatively solve one variable such as $\boldsymbol{\alpha}$ given the other variables such as $\mathbf{D}$ and $\boldsymbol{\lambda}$ fixed.

3.4.1 Compute $\boldsymbol{\alpha}$ when $\boldsymbol{\lambda}$ and $\mathbf{D}$ are fixed

Fixing $\mathbf{D}$ and $\boldsymbol{\lambda}$, the optimization problem in Eq.(18) reduces to

$$\begin{aligned}\min_{\boldsymbol{\alpha}} \quad & g(\boldsymbol{\alpha})\\ \text{s.t.} \quad & 0\le\alpha_{ikl}\le\frac{C}{|y_i||\bar{y}_i|},\end{aligned} \tag{19}$$

where $g(\boldsymbol{\alpha})$ is defined as

$$\begin{aligned}g(\boldsymbol{\alpha}) &= \frac{1}{2}\sum_{k,l=1}^{c}R_{kl}^{-1}\sum_{i,j=1}^{n}\beta_{ik}\beta_{jl}(\bar{\mathbf{p}}\circ\bar{\mathbf{x}}_i)^T(\bar{\mathbf{p}}\circ\bar{\mathbf{x}}_j)\\ &- \sum_{i=1}^{n}\sum_{(k,l)\in y_i\times\bar{y}_i}\alpha_{ikl},\end{aligned} \tag{20}$$

where $\bar{\mathbf{p}}=[\mathbf{p}_1^T,\ldots,\mathbf{p}_{|\mathcal{P}|}^T]^T$ and $\bar{\mathbf{x}}_i=[\lambda_1\mathbf{x}_i,\ldots,\lambda_{|\mathcal{P}|}\mathbf{x}_i^T]^T$.

The above optimization problem can be efficiently solved by dual coordinate descent method [12]. It updates one variable at a time by minimizing a single variable subproblem. In particular, it picks one variable α_{ikl} at a time and solves the following single variable subproblem while keeping all the other variables fixed,

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & g(\boldsymbol{\alpha} + d\mathbf{e}_{ikl}), \\ \text{s.t.} \quad & 0 \leq \alpha_{ikl} \leq \frac{C}{|y_i||\bar{y}_i|}, \end{aligned} \tag{21}$$

where $\mathbf{e}_{ikl} = (0, \ldots, 0, 1, 0, \ldots, 0)^T$. The objective function of Eq.(21) is a simple quadratic function of d,

$$\begin{aligned} g(\boldsymbol{\alpha} + d\mathbf{e}_{ikl}) &= \frac{S_{kl}}{2}(\bar{\mathbf{p}} \circ \bar{\mathbf{x}}_i)^T(\bar{\mathbf{p}} \circ \bar{\mathbf{x}}_i)d^2 \\ &+ \nabla_{ikl} g(\boldsymbol{\alpha})d + \text{const}, \end{aligned} \tag{22}$$

where $S_{kl} = R_{kk}^{-1} + R_{ll}^{-1} - 2R_{kl}^{-1}$, const is a constant which is independent on d, and $\nabla_{ikl} g(\boldsymbol{\alpha})$ can be computed as

$$\begin{aligned} & \nabla_{ikl} g(\boldsymbol{\alpha}) \\ = & \sum_{q=1}^{c}(R_{kq}^{-1} - R_{lq}^{-1}) \sum_{j=1}^{n} \beta_{jq}(\bar{\mathbf{p}} \circ \bar{\mathbf{x}}_j)^T(\bar{\mathbf{p}} \circ \bar{\mathbf{x}}_i) - 1 \\ = & \sum_{q=1}^{c}(R_{kq}^{-1} - R_{lq}^{-1})(\sum_{l=1}^{c} R_{ql}\mathbf{w}_l)^T(\bar{\mathbf{p}} \circ \bar{\mathbf{x}}_i) - 1 \\ = & \sum_{q=1}^{c}(R_{kq}^{-1} - R_{lq}^{-1})\mathbf{u}_q^T(\bar{\mathbf{p}} \circ \bar{\mathbf{x}}_i) - 1 \\ = & (\mathbf{w}_k - \mathbf{w}_l)^T(\bar{\mathbf{p}} \circ \bar{\mathbf{x}}_i) - 1, \end{aligned} \tag{23}$$

where $\mathbf{u}_k = \sum_{l=1}^{c} R_{kl}\mathbf{w}_l$. It can be easily seen that Eq.(21) has an optimum at $d = 0$ if and only if

$$\nabla_{ikl}^P g(\boldsymbol{\alpha}) = 0, \tag{24}$$

where $\frac{\partial g(\boldsymbol{\alpha})}{\partial \alpha_{ikl}}$ is the projected gradient

$$\nabla_{ikl}^P g(\boldsymbol{\alpha}) = \begin{cases} \nabla_{ikl} g(\boldsymbol{\alpha}), & \text{if } 0 < \alpha_{ikl} < \frac{C}{|y_i||\bar{y}_i|} \\ \min(0, \nabla_{ikl} g(\boldsymbol{\alpha})), & \text{if } \alpha_{ikl} = 0 \\ \max(0, \nabla_{ikl} g(\boldsymbol{\alpha})), & \text{if } \alpha_{ikl} = \frac{C}{|y_i||\bar{y}_i|} \end{cases}. \tag{25}$$

If Eq.(25) holds, we do not need to update α_{ikl} and directly move to the next variable. Otherwise, the optimal solution of Eq.(21) is

$$\alpha_{ikl}^* = \min(\max(\alpha_{ikl} - \frac{\nabla_{ikl}^P g(\boldsymbol{\alpha})}{S_{kl}(\bar{\mathbf{p}} \circ \bar{\mathbf{x}}_i)^T(\bar{\mathbf{p}} \circ \bar{\mathbf{x}}_i)}, 0), \frac{C}{|y_i||\bar{y}_i|}). \tag{26}$$

This means the subproblem can be solved analytically that ensures the efficiency of the coordinate descent method. Here, we need to calculate $(\bar{\mathbf{p}} \circ \bar{\mathbf{x}}_i)^T(\bar{\mathbf{p}} \circ \bar{\mathbf{x}}_i)$ and $\nabla_{ikl}^P g(\boldsymbol{\alpha})$. First, $(\bar{\mathbf{p}} \circ \bar{\mathbf{x}}_i)^T(\bar{\mathbf{p}} \circ \bar{\mathbf{x}}_i)$ can be pre-computed and stored in the memory. Second, to evaluate $\nabla_{ikl}^P g(\boldsymbol{\alpha})$ using Eq.(23), we only need to maintain $\mathbf{u}_k$ by

$$\begin{aligned} \mathbf{u}_k &= \mathbf{u}_k + (\alpha_{ikl}^* - \alpha_{ikl})\bar{\mathbf{p}} \circ \bar{\mathbf{x}}_i \\ \mathbf{u}_l &= \mathbf{u}_l - (\alpha_{ikl}^* - \alpha_{ikl})\bar{\mathbf{p}} \circ \bar{\mathbf{x}}_i. \end{aligned} \tag{27}$$

The dual coordinate descent method for optimizing $\boldsymbol{\alpha}$ is summarized in Algorithm 1.

Theorem 3.3 *The $\boldsymbol{\alpha}$ calculated by Algorithm 1 globally converges to an optimal solution $\boldsymbol{\alpha}^*$. The convergence rate is at least linear. In other words, there is $0 < \tau < 1$ and an iteration t_0, such that*

$$g(\boldsymbol{\alpha}^{t+1}) - g(\boldsymbol{\alpha}^*) \leq \tau(g(\boldsymbol{\alpha}^t) - g(\boldsymbol{\alpha}^*)). \tag{28}$$

PROOF. Please refer to [12]. □

Algorithm 1 Dual Coordinate Descent for Optimizing $\boldsymbol{\alpha}$

Input: C and m;
Output: $\boldsymbol{\alpha}$;
Initialize $\boldsymbol{\alpha} = 0$ and $\mathbf{w}_k = 0, k = 1, \ldots, c$;
repeat
  for $i = 1, \ldots, n$ and $(k, l) \in y_i \times \bar{y}_i$ **do**
    Calculate $G = R_{kl}^{-1}(\mathbf{u}_k - \mathbf{u}_l)^T\mathbf{x}_i - 1$;
    Calculate $PG = \begin{cases} G, & \text{if } 0 < \alpha_{ikl} < \frac{C}{|y_i||\bar{y}_i|} \\ \min(0, G), & \text{if } \alpha_{ikl} = 0 \\ \max(0, G), & \text{if } \alpha_{ikl} = \frac{C}{|y_i||\bar{y}_i|} \end{cases}$.
    if $|PG| \neq 0$ **then**
      $\alpha_{ikl}^* = \min(\max(\alpha_{ikl} - \frac{PG}{S_{kl}(\bar{\mathbf{p}} \circ \bar{\mathbf{x}}_i)^T(\bar{\mathbf{p}} \circ \bar{\mathbf{x}}_i)}, 0), \frac{C}{|y_i||\bar{y}_i|})$
      Calculate $\mathbf{u}_k = \mathbf{u}_k + (\alpha_{ikl}^* - \alpha_{ikl})\bar{\mathbf{p}} \circ \bar{\mathbf{x}}_i$
      Calculate $\mathbf{u}_l = \mathbf{u}_l - (\alpha_{ikl}^* - \alpha_{ikl})\bar{\mathbf{p}} \circ \bar{\mathbf{x}}_i$
    end if
  end for
until converge

The linear convergence result is remarkable, that means Algorithm can achieves an ϵ-accurate solution $\boldsymbol{\alpha}$ in $O(\log(\frac{1}{\epsilon}))$ iterations.

3.4.2 Compute $\mathbf{D}$ when $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$ are fixed

Given $\boldsymbol{\alpha}$ and $\boldsymbol{\lambda}$, the optimization problem in Eq.(6) boils down to

$$\begin{aligned} \min_{\mathbf{D}} \quad & \text{tr}(\mathbf{W}\mathbf{D}^{-1}\mathbf{W}^T), \\ s.t. \quad & \mathbf{D} \succeq 0, \text{tr}(\mathbf{D}) = 1, \end{aligned} \tag{29}$$

which is a semi-definite programming (SDP) [2]. Fortunately, it can be solved by spectral method as stated in the following theorem.

Theorem 3.4 *Let $\mathbf{C} = \mathbf{W}^T\mathbf{W}$, the optimal solution of Eq.(29) is*

$$\mathbf{D} = \frac{\mathbf{C}^{\frac{1}{2}}}{tr(\mathbf{C}^{\frac{1}{2}})}, \tag{30}$$

and the optimal value equals to $(tr(\mathbf{C}^{\frac{1}{2}}))^2$

PROOF. The proof is similar to Theorem 4.6 in [9]. Let $\mathbf{D} = \mathbf{A}\text{diag}(\boldsymbol{\lambda})\mathbf{A}^T$ where $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_c] \in \mathbb{R}^d$, then

$$\begin{aligned} & \sum_{j=1}^{d} \mathbf{w}^j\mathbf{D}^{-1}(\mathbf{w}^j)^T = \text{tr}(\mathbf{W}\mathbf{D}^{-1}\mathbf{W}^T) \\ = & \text{tr}(\mathbf{W}\mathbf{A}\text{diag}(\boldsymbol{\lambda})^{-1}\mathbf{A}^T\mathbf{W}^T) \\ = & \text{tr}(\text{diag}(\lambda)^{-1}\mathbf{A}^T\mathbf{W}^T\mathbf{W}\mathbf{A}) \\ = & \sum_{k=1}^{c} \frac{\mathbf{a}_k^T\mathbf{W}^T\mathbf{W}\mathbf{a}_k}{\lambda_k} \geq (\sum_{k=1}^{c} ||\mathbf{W}\mathbf{a}_k||_2)^2 \end{aligned}$$

Next, we have

$$\begin{aligned} & ||\mathbf{W}\mathbf{a}_k||_2^2 = \mathbf{a}_k^T\mathbf{W}^T\mathbf{W}\mathbf{a}_k \\ = \ & \mathbf{a}_k^T\mathbf{C}\mathbf{a}_k = (\mathbf{a}_k^T\mathbf{C}\mathbf{a}_k)(\mathbf{a}_k^T\mathbf{a}_k) \\ = \ & \mathrm{tr}(\mathbf{C}^{\frac{1}{2}}\mathbf{a}_k\mathbf{a}_k^T\mathbf{C}^{\frac{1}{2}})\mathrm{tr}(\mathbf{a}_k\mathbf{a}_k^T) \\ \geq \ & \mathrm{tr}(\mathbf{C}^{\frac{1}{2}}\mathbf{a}_k\mathbf{a}_k^T\mathbf{C}^{\frac{1}{2}}\mathbf{a}_k\mathbf{a}_k^T) \\ = \ & \mathrm{tr}(\mathbf{a}_k^T\mathbf{C}^{\frac{1}{2}}\mathbf{a}_k\mathbf{a}_k^T\mathbf{C}^{\frac{1}{2}}\mathbf{a}_k) = (\mathbf{a}_k^T\mathbf{C}^{\frac{1}{2}}\mathbf{a}_k)^2 \end{aligned}$$

since $\mathrm{tr}(\mathbf{A})\mathrm{tr}(\mathbf{B}) \geq \mathrm{tr}(\mathbf{AB})$ if $\mathbf{A}$ and $\mathbf{B}$ are positive semi-definite. The equality holds if and only if $\mathbf{C}^{\frac{1}{2}}\mathbf{a}_k\mathbf{a}_k^T = \mu\mathbf{a}_k\mathbf{a}_k^T$ which implies that $\mathbf{C}^{\frac{1}{2}}\mathbf{a}_k = \mu\mathbf{a}_k$, that is, $\mathbf{a}_k$ is an eigenvector of $\mathbf{C}^{\frac{1}{2}}$. The optimal μ is $\mathrm{tr}(C^{\frac{1}{2}})$. Hence we obtain

$$\begin{aligned} & \sum_{j=1}^{d}\mathbf{w}^j\mathbf{D}^{-1}(\mathbf{w}^j)^T \geq (\sum_{k=1}^{c}\mathbf{a}_k^T\mathbf{C}^{\frac{1}{2}}\mathbf{a}_k)^2 \\ = \ & (\mathrm{tr}(\mathbf{A}^T\mathbf{C}^{\frac{1}{2}}\mathbf{A}))^2 = (\mathrm{tr}(\mathbf{C}^{\frac{1}{2}}))^2 \end{aligned}$$

Consequently, the optimal $\mathbf{D} = \frac{\mathbf{C}^{\frac{1}{2}}}{\mathrm{tr}(\mathbf{C}^{\frac{1}{2}})}$. This completes the proof. □

3.4.3 Compute $\boldsymbol{\lambda}$ when $\boldsymbol{\alpha}$ and $\mathbf{D}$ are fixed

Let $h(\boldsymbol{\lambda}) = \sum_t \lambda_t f(\boldsymbol{\alpha}, \mathbf{D}, \mathbf{p}^t)$, we denote the sub-gradient of $h(\boldsymbol{\lambda})$ with respect to λ_t by $\nabla_{\lambda_t}h(\boldsymbol{\lambda})$, which is calculated as

$$\nabla_{\lambda_t}h(\boldsymbol{\lambda}) = -\frac{1}{2}\sum_{k,l=1}^{c}R_{kl}^{-1}\sum_{i,j=1}^{n}\beta_{ik}\beta_{jl}(\mathbf{p}^t\circ\mathbf{x}_i)^T(\mathbf{p}^t\circ\mathbf{x}_j). \quad (31)$$

Following [22], we use projected gradient descent to update the kernel weights λ_t. Note that other techniques such as semi-infinite linear programming [26] and extended level method [32] can also be adopted.

3.5 Cutting Plane Acceleration

Up to now, we have presented the algorithm for optimizing Eq.(18). However, given $\mathcal{P}$, the problem has optimization variables $(\boldsymbol{\alpha}, \mathbf{D}, \boldsymbol{\lambda})$ with $\binom{d}{m}$ constraints, which is impractical to solve. Fortunately, cutting plane technique [17] enables us to deal with this problem, which keeps a polynomial sized subset Ω of working constraints and computes the optimal solution to Eq. (18) subject to the constraints in Ω. In detail, the algorithm adds the most violated constraint in Eq. (15) into Ω in each iteration. In this way, a successively strengthening approximation of the original problem is solved. And the algorithm terminates when no constraints in Eq. (15) is violated.

The remaining thing is how to find the most violated constraint in each iteration. Since the feasibility of a constraint is measured by the corresponding value of θ, the most violated constraint is the one which owns the largest θ. Hence, it could be calculated as follows

$$\begin{aligned} & \arg\max_{\mathbf{p}\in\mathcal{P}} -f(\boldsymbol{\alpha}, \mathbf{D}, \mathbf{p}) \\ = \ & \arg\max_{\mathbf{p}\in\mathcal{P}} \sum_{k,l=1}^{c}R_{kl}\sum_{i,j=1}^{n}\beta_{ik}\beta_{jl}(\mathbf{p}\circ\mathbf{x}_i)^T(\mathbf{p}\circ\mathbf{x}_j) \\ = \ & \arg\max_{\mathbf{p}\in\mathcal{P}} \sum_{j=1}^{d}s_j p_j, \end{aligned} \quad (32)$$

where $s_j = \sum_{k,l=1}^{c}R_{kl}\boldsymbol{\beta}_k^T(\mathbf{x}^j)^T\mathbf{x}^j\boldsymbol{\beta}_l$ and $\boldsymbol{\beta}_k = [\beta_{1k}, \ldots, \beta_{nk}]^T$. According to [28], its optimal solution can be obtained without any numeric optimization solver. Instead, it can be solved by first sorting s_j and then setting the first m numbers corresponding to d_j to 1 and the rests to 0.

We summarize the algorithm to solve the problem in Eq. (18) in Algorithm 2. Note that the final selected features are the union set of the features corresponding to each constraint $\mathbf{p}^t \in \Omega_T$.

Algorithm 2 Correlated Multi-Label Feature Selection

Input: C and m;
Output: $\boldsymbol{\alpha}$ and Ω;
Initialize $\boldsymbol{\alpha} = \mathbf{0}$ and $t = 1$;
Find the most violated constraint $\mathbf{p}^1$, and set $\Omega_1 = \{\mathbf{p}^1\}$;
repeat
  Initialize $\boldsymbol{\lambda} = \frac{1}{t}\mathbf{1}$;
  repeat
    Solve for $\boldsymbol{\alpha}$ using Algorithm 1;
    Solve for $\mathbf{D}$ using Eq.(30);
    Solve for $\boldsymbol{\lambda}$ using sub-gradient descent as in Eq. (31);
  until converge
  Find the most violated constraint $\mathbf{p}^{t+1}$ and set $\Omega_{t+1} = \Omega_t \cup \mathbf{p}^{t+1}$;
  $t = t + 1$;
until converge

3.6 Convergence Analysis

We analyze the convergence property of Algorithm 2.

Theorem 3.5 *Let $(\alpha^*, \mathbf{D}^*, \theta^*)$ be the global optimal solution of Eq. (15), $l_t = \max_{1\leq j\leq t}\min_{\boldsymbol{\alpha}\in\mathcal{A},\mathbf{D}\in\mathcal{D},} -f(\boldsymbol{\alpha}, \mathbf{D}, \mathbf{p}^j)$ and $u_t = \min_{1\leq j\leq t}\max_{\mathbf{p}\in\mathcal{P}} -f(\boldsymbol{\alpha}^j, \mathbf{D}^j, \mathbf{p})$, then*

$$l_t \leq \theta^* \leq u_t. \quad (33)$$

With the number of iteration t increasing, the sequence $\{l_t\}$ is monotonically increasing and the sequence $\{u_t\}$ is monotonically decreasing.

Proof. Please refer to [28]. □

Since the number of constraints in $\mathcal{P}$ is finite, i.e., $\binom{d}{m}$, based on Theorem 3.5, the algorithm will converge within finite number of iterations. Moreover, we can use the gap between l_t and u_t to trace the convergence of Algorithm 2. When the gap is smaller than a predefined tolerance ϵ, we stop the algorithm. Empirical study shows that the algorithm converges within 10 outer-iterations in our experiments.

3.7 Time Complexity Analysis

In each outer iteration of Algorithm 2, it needs to find the most violated $\mathbf{p}$. It can be obtained exactly by finding the m largest ones from d coefficients s_j, which takes only $O(m\log d)$ time. In the inner iteration of Algorithm 2, it solves a minimax problem by alternating optimization. Its complexity is proportional to the sum of the complexity of dual coordinate descent and the complexity of calculating $\mathbf{D}$. The complexity of dual coordinate descent is $O(c^2ns)$, where s is the average number of nonzero features among all the training samples. The complexity of calculating $\mathbf{D}$ is $O(dc)$. Hence the total time complexity of the proposed method is $O(T(c^2ns + dc + m\log d))$, where T is the number

of iterations needed to converge, Thus, the proposed method is computationally efficient for large-scale, high dimensional data.

3.8 Set Size Prediction

So far we have only developed a ranking system. To obtain the final labels of each instance, we need to design a label set size predictor $s(\mathbf{x})$. Following [7], we turn to learn a threshold function $t(\mathbf{x})$, which differentiates labels in the target set from others. Given the threshold function, the predictor of the set size is quite straightforward: $s(\mathbf{x}) = |\{k|\mathbf{w}_k^T\mathbf{x} > t(\mathbf{x})\}|$. The remaining problem is how to learn $t(\mathbf{x})$. We formulate it as a regression problem. In detail, given a training instance $\mathbf{x}_i$, its label ranking scores are $\mathbf{w}_1^T\mathbf{x}, \ldots, \mathbf{w}_c^T\mathbf{x}$, we define the corresponding threshold $t_i = t(\mathbf{x}_i)$ by

$$t(\mathbf{x}_i) = \frac{1}{2}(\min_{k\in y_i} \mathbf{w}_k^T\mathbf{x}_i + \max_{l\in \bar{y}_i} \mathbf{w}_l^T\mathbf{x}_i) \tag{34}$$

Once we generate the thresholds $\{t_i = t(\mathbf{x}_i)\}_{i=1}^n$ for the training set, we can estimate the threshold function $t(\mathbf{x})$ by any regression model. In this paper, we simply use linear regression to estimate $t(\mathbf{x})$.

4. EXPERIMENTS

In this section, we empirically evaluate the effectiveness of the proposed methods. All experiments are performed on a PC with Intel Core i5 3.20G CPU and 4GB RAM and all algorithms in our experiments are implemented in Matlab and C++.

4.1 Data Sets

We carry our experiments on various sets of data, including data sets from LibSVM website[2] and Yahoo[3]. In LibSVM data sets, we choose scene data set [1] and yeast data set [7]. For Yahoo data set [30], we use four categories: Arts, Business, Education, and Health as four data sets. In each category, the sub-categories are the labels for each document. We pre-processed the data sets by removing sub-categories with less than 100 documents and documents with no sub-category. Table 1 summarizes the characteristics of these data sets.

Table 1: Description of the data sets

Datasets	#training	#features	#classes
Scene	2407	294	6
Yeast	2417	103	14
Arts	7441	17973	19
Business	9968	16621	17
Education	11817	20782	14
Health	9109	18430	14

4.2 Evaluation Metrics

To evaluate the performance of different algorithms for multi-label learning, we use three measures: the Area Under the Receiver Operating Characteristic (ROC) Curve (AUC), Micro F1 and Macro F1. For AUC, we first compute the AUC for each class, and then compute the averaged AUC over all the classes. For more details about the measures, please refer to [33].

4.3 Parameter Settings

We compare the proposed methods with the related multi-label learning methods and a single-label feature selection method. We choose Fisher score [21] as the representative of single-label feature selection methods. The reason is that our empirical study [8] found that Fisher score is comparable to or even better than the other feature selection methods [34, 24] on the data sets used in our experiments. We will not report the results of MDDM [39] and MLDA [31] because they are not better than MLLS [13]. All the methods and their parameter settings are summarized as follows. By default, the regularization parameter C of SVM type models in all the methods is tune by 5-fold cross validation on the training set via searching the grid $\{10^{-3}, 10^{-2}, \ldots, 10^3\}$.

SVM: linear SVM is used for one-against-all classification individually.

LaRank SVM [7]: The threshold function is learned by linear regression as stated in Section 3.8.

CCA+SVM: Canonical Correlation Analysis is used for dimensionality reduction before SVM. The dimensionality of the subspace is set to $c-1$ where c is the number of classes. The regularization parameter for CCA is tuned by 5-fold cross validation on the training set via searching the grid $\{10^{-5}, 10^{-4}, \ldots, 10^{-1}\}$.

MLLS[4] [13]: The dimensionality of the subspace is set to $c-1$. The two regularization parameters for MLLS is tuned by 5-fold cross validation on the training set by searching the grid $\{10^{-5}, 10^{-4}, \ldots, 10^{-1}\}$.

FS+SVM [21]: Fisher score (FS) is used for each one-against-all binary classification individually, followed with linear SVM. The number of selected features m is tuned by 5-fold cross validation via searching the grid $\{10, 20, \ldots, \lfloor \frac{n}{10} \rfloor \times 10\}$ on Scene and Yeast data sets, and via the grid $\{1000, 2000, \ldots, \lfloor \frac{n}{1000} \rfloor \times 1000\}$ on the Yahoo data sets.

CLaRank SVM: The regularization parameter μ is tuned by 5-fold cross validation on the training set by searching the grid $\{10^{-2}, 10^{-1}, \ldots, 10^2\}$.

MLFS: The m which controls the number of features is also tuned by 5-fold cross validation on the training set over the grid $\{10, 11, \ldots, 20\}$ on scene and yeast data sets, and over the grid $\{1000, 2000, \ldots, 10000\}$ on Yahoo data sets.

CMLFS: The m is tuned the same as above. The regularization parameter μ is tuned the same as CLaRank SVM.

Since learning with small number of labelled data is much more challenging than learning with large number of labelled data, we randomly sample 1000 data points from each data set for training, and the rest for testing. Note that each label is guaranteed to appear in at least one data point of the training set and in at least one data point of the testing set. The process was repeated 10 times and the mean along with standard deviation of measures are reported.

4.4 Classification Results

The classification results of all the methods are shown in Table 2. CMLFS, MLFS and CLaRank SVM are the three methods proposed by us. As is pointed out before, MLFS and CLaRank SVM are two special instances of CMLFS.

[2]http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/multilabel.html

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

[4]http://www.public.asu.edu/s̃ji03/multilabel/

Compared with other methods, these three methods show the best performance. Specifically, CLaRank SVM achieves better results because it considers label correlation. And the good performance of MLFS takes advantage of feature selection. Among these three, CMLFS performs the best. This demonstrates that it is very necessary to combine label correlation and feature selection into classification model simultaneously.

Two state-of-the-art methods, LaRank SVM and MLLS, perform worse than our methods but still are better than the others. LaRank SVM classifies (ranks) all the labels simultaneously. It considers the ordinal (first-order) information between labels, which is beneficial for multi-label learning. MLLS is based on subspace learning, it is able to learn a discriminative subspace for multi-label classification. That is why MLLS generally shows better results than LaRank SVM on Yahoo data sets, which are of high dimensionality.

Let us take a closer look at LaRank SVM in comparison with the proposed methods. CLaRank SVM improves LaRank SVM consistently on all the data sets. The improvement is a result of using the correlation (second-order) information among labels. As we mentioned above, LaRank SVM only considers the first-order information. On the other hand, MLFS performs better than LaRank SVM. The reason is that the feature selection in MLFS helps it avoid the curse of dimensionality.

Comparing MLLS with MLFS, the major problem of MLLS is that it fails to consider the label correlation. Though both MLLS and MLFS involve dimensionality reduction, MLFS is superior to MLLS at most cases because it considers the label rank (first-order information).

CCA+SVM and FS+SVM fail to perform well because both of them are in the fashion of two-stage approach. For CCA+SVM, it first carries out subspace learning and then learns a classification model. FS+SVM does feature selection followed by learning the classifier. Both methods try to deal with high dimensional data but fail to integrate dimensionality reduction and classifier learning into a unified framework. Besides, Fisher score is used in each one-against-all binary classification independently, so the selected features are generally different from each other for each binary classification problem.

4.5 AUC v.s. Number of Features

In this subsection, we study the performance of multi-label learning with respect to the number of selected features. We compare CMLFS and MLFS with the single-label feature selection method, i.e., Fisher score. Since the number of selected features for the CMLFS and MLFS is determined implicitly by m, we increase m gradually and obtain a increasing number of features. Figure 2 depicts the AUC with respect to the increasing number of selected features.

We can see that with a very small number of features, MLFS and CMLFS can achieve very good performance. In contrast, the performance of Fisher score is pretty bad. In fact, the satisfying classification results of Fisher score shown in Table 2 are achieved by selecting almost all the features. This again strengthens the superiorness of the proposed multi-label feature selection methods over single-label feature selection method.

5. CONCLUSION AND FUTURE WORK

In this paper, we present a multi-label feature selection method based on LaRank SVM. It is formulated as quadratically constrained linear programming and solved by cutting plane algorithm, in each iteration of which a minimax optimization problem is solved by dual coordinate descent and stochastic sub-gradient descent alternatively. Its training time is linear in the number of training samples, which enables it applicable to large scale multi-label data.

In our future work, we will study how to solve the feature selection problem in the primal [15, 25]. Moreover, we also plan to study semi-supervised multi-label learning [3].

Acknowledgements

The work was supported in part by NSF IIS-09-05215, U.S. Air Force Office of Scientific Research MURI award FA9550-08-1-0265, and the U.S. Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (NS-CTA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on. We thank the anonymous reviewers for their helpful comments.

6. REFERENCES

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

[2] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, 2004.

[3] G. Chen, Y. Song, F. Wang, and C. Zhang. Semi-supervised multi-label learning by solving a sylvester equation. In *SDM*, pages 410–419, 2008.

[4] G. Chen, J. Zhang, F. Wang, C. Zhang, and Y. Gao. Efficient multi-label classification with hypergraph regularization. In *CVPR*, pages 1658–1665, 2009.

[5] J. Chen and J. Ye. Training svm with indefinite kernels. In *ICML*, pages 136–143, 2008.

[6] O. Dekel, C. D. Manning, and Y. Singer. Log-linear models for label ranking. In *NIPS*, 2003.

[7] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *NIPS*, pages 681–687, 2001.

[8] Q. Gu, Z. Li, and J. Han. Generalized fisher score for feature selection. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, 2011.

[9] Q. Gu and J. Zhou. Subspace maximum margin clustering. In *CIKM*, pages 1337–1346, 2009.

[10] A. K. Gupta and D. K. Nagar. *Matrix Variate Distributions*, volume 104 of *Monographs and Surveys in Pure and Applied Mathematics*. Chapman Hall/CRC, Florida, 2000.

[11] B. Hariharan, L. Zelnik-Manor, S. V. N. Vishwanathan, and M. Varma. Large scale max-margin multi-label classification with priors. In *ICML*, pages 423–430, 2010.

[12] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear svm. In *ICML*, pages 408–415, 2008.

Table 2: The classification results (AUC, Micro F1 and Macro F1) on the six data sets. The higher the measures are, the better the performance is.

Method	Measure	Scene	Yeast	Arts	Business	Education	Health
SVM	AUC	91.60±0.20	65.63±0.89	73.31±0.38	81.31±1.12	75.15±0.63	82.97±0.33
	Micro F1	68.45±0.84	60.52±0.52	45.47±0.90	76.62±0.46	48.32±1.11	64.89±0.64
	Macro F1	69.78±0.91	46.32±0.64	33.64±0.90	40.37±2.74	38.47±0.92	51.31±1.21
LaRank SVM	AUC	91.85±0.47	66.35±0.58	75.94±0.47	81.38±0.94	77.37±0.72	85.25±0.33
	Micro F1	68.67±1.00	61.55±0.71	46.61±1.28	76.76±1.67	49.68±1.53	67.33±1.23
	Macro F1	69.04±0.87	47.14±0.44	33.67±0.85	40.45±2.29	38.50±2.11	59.51±1.36
CCA+SVM	AUC	85.41±0.49	64.40±1.04	72.18±0.27	80.46±0.98	73.61±0.65	83.02±0.48
	Micro F1	58.06±0.69	59.80±0.98	40.01±1.86	70.77±5.58	42.74±3.17	62.76±1.39
	Macro F1	60.24±0.82	45.28±0.84	28.72±0.95	35.92±2.70	34.10±1.62	53.52±2.07
MLLS	AUC	91.67±0.23	67.82±0.63	77.15±0.39	83.53±0.69	78.09±0.61	86.37±0.29
	Micro F1	68.93±0.97	**62.60±0.26**	46.70±0.71	76.52±0.59	49.42±0.78	68.53±0.48
	Macro F1	70.09±0.90	46.60±0.68	34.61±0.91	41.09±2.12	39.39±1.50	60.07±0.98
FS+SVM	AUC	91.59±0.20	65.92±0.75	73.45±0.45	81.31±1.12	75.15±0.63	84.57±0.45
	Micro F1	68.48±1.07	60.50±0.75	43.70±1.10	76.62±0.46	48.32±1.11	67.49±0.72
	Macro F1	69.72±1.07	46.41±0.72	31.25±0.92	40.37±2.74	38.47±0.92	58.99±1.01
CLaRank SVM	AUC	92.06±0.20	67.39±0.62	76.16±0.47	82.39±0.71	78.52±0.73	86.43±0.49
	Micro F1	69.10±0.50	61.52±0.73	46.89±1.32	76.84±1.12	49.37±1.16	67.79±1.34
	Macro F1	**71.51±0.58**	47.27±0.36	34.50±0.40	40.50±2.09	39.75±1.17	60.93±2.17
MLFS	AUC	91.80±0.24	67.25±0.94	78.69±0.24	84.70±0.64	76.44±0.50	88.20±0.33
	Micro F1	68.39±1.80	60.14±1.11	47.24±0.95	77.54±1.27	48.76±1.31	69.34±1.26
	Macro F1	69.91±0.85	47.21±0.71	34.64±0.72	40.94±1.81	38.43±1.06	61.16±1.87
CMLFS	AUC	**93.11±0.41**	**68.68±1.07**	**79.88±0.23**	**85.81±0.64**	**80.52±0.52**	**89.28±0.33**
	Micro F1	**69.80±1.82**	62.55±0.88	**47.83±1.25**	**77.04±1.17**	**50.04±1.62**	**69.39±1.49**
	Macro F1	70.43±0.57	**47.30±0.77**	**35.42±0.67**	**41.65±1.61**	**40.85±0.96**	**62.10±1.47**

[13] S. Ji, L. Tang, S. Yu, and J. Ye. Extracting shared subspace for multi-label classification. In *KDD*, pages 381–389, 2008.

[14] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, pages 133–142, 2002.

[15] T. Joachims. Training linear svms in linear time. In *KDD*, pages 217–226, 2006.

[16] T. Joachims, T. Finley, and C.-N. J. Yu. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, 2009.

[17] J. E. Kelley. The cutting plane method for solving convex programs. *Journal of the SIAM*, 8:703–712, 1960.

[18] S.-J. Kim and S. Boyd. A minimax theorem with applications to machine learning, signal processing, and finance. *SIAM J. on Optimization*, 19:1344–1367, November 2008.

[19] 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.

[20] Y. Li, I. Tsang, J. Kwok, and Z. Zhou. Tighter and convex maximum margin clusteringe. In *AISTATS*, 2009.

[21] P. E. H. R. O. Duda and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2001.

[22] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. More efficiency in multiple kernel learning. In *ICML*, pages 775–782, 2007.

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

[24] M. Rogati and Y. Yang. High-performing feature selection for text classification. In *CIKM*, pages 659–661, 2002.

[25] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *ICML*, pages 807–814, 2007.

[26] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

[27] L. Sun, S. Ji, and J. Ye. Hypergraph spectral learning for multi-label classification. In *KDD*, pages 668–676, 2008.

[28] M. Tan, L. Wang, and I. W. Tsang. Learning sparse svm for feature selection on very high dimensional datasets. In *ICML*, pages 1047–1054, 2010.

[29] L. Tang, J. Chen, and J. Ye. On multiple kernel learning with multiple labels. In *IJCAI*, pages 1255–1260, 2009.

[30] N. Ueda and K. Saito. Parametric mixture models for multi-labeled text. In *NIPS*, pages 721–728, 2002.

[31] H. Wang, C. H. Q. Ding, and H. Huang. Multi-label linear discriminant analysis. In *ECCV (6)*, pages 126–139, 2010.

[32] Z. Xu, R. Jin, I. King, and M. R. Lyu. An extended level method for efficient multiple kernel learning. In *NIPS*, pages 1825–1832, 2008.

[33] Y. Yang. An evaluation of statistical approaches to text categorization. *Inf. Retr.*, 1(1-2):69–90, 1999.

[34] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, pages 412–420, 1997.

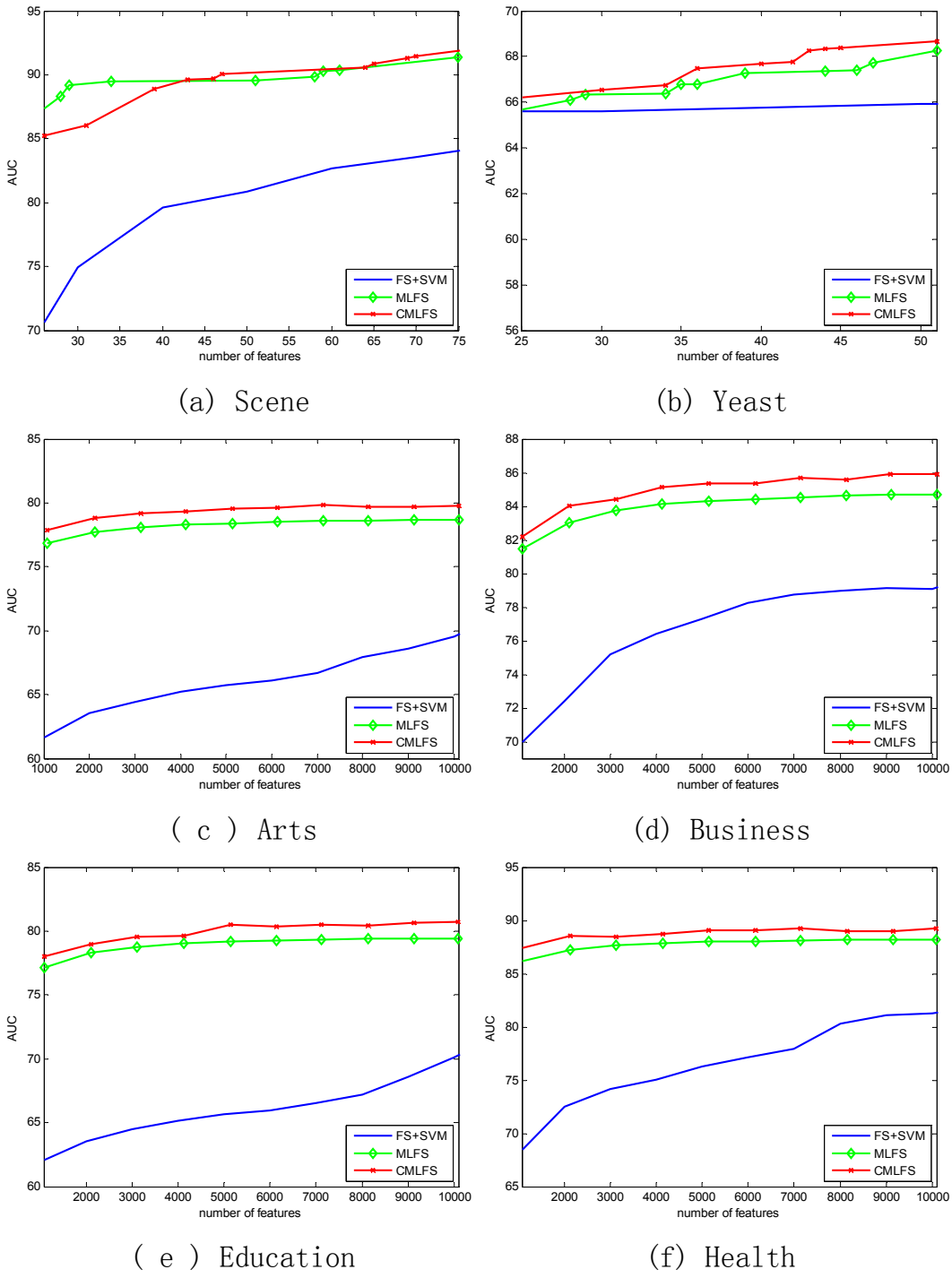

Figure 2: AUC of FS+SVM, MLFS and CMLFS with respect to the number of features on the six data sets

[35] K. Yu, S. Yu, and V. Tresp. Multi-label informed latent semantic indexing. In *SIGIR*, pages 258–265, 2005.

[36] M.-L. Zhang and K. Zhang. Multi-label learning by exploiting label dependency. In *KDD*, pages 999–1008, 2010.

[37] Y. Zhang and D. yan Yeung. A convex formulation for learning task relationships in multi-task learning. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2010.

[38] Y. Zhang and D.-Y. Yeung. Transfer metric learning by learning task relationships. In *KDD*, pages 1199–1208, 2010.

[39] Y. Zhang and Z.-H. Zhou. Multi-label dimensionality reduction via dependence maximization. In *AAAI*, pages 1503–1505, 2008.