Semi-supervised Multi-label Learning by Solving a Sylvester Equation

Gang Chen * Yangqiu Song* Fei Wang* Changshui Zhang*

Abstract

Multi-label learning refers to the problems where an instance can be assigned to more than one category. In this paper, we present a novel Semi-supervised algorithm for Multi-label learning by solving a *Sylvester Equation* (*SMSE*). Two graphs are first constructed on *instance* level and *category* level respectively. For *instance* level, a graph is defined based on both labeled and unlabeled instances, where each node represents one instance and each edge weight reflects the similarity between corresponding pairwise instances. Similarly, for *category* level, a graph is also built based on all the categories, where each node represents one category and each edge weight reflects the similarity between corresponding pairwise categories. A regularization framework combining two regularization terms for the two graphs is suggested. The regularization term for *instance* graph measures the smoothness of the labels of instances, and the regularization term for *category* graph measures the smoothness of the labels of categories. We show that the labels of unlabeled data finally can be obtained by solving a *Sylvester Equation*. Experiments on *RCV1* data set show that *SMSE* can make full use of the unlabeled data information as well as the correlations among categories and achieve good performance. In addition, we give a *SMSE*'s extended application on collaborative filtering.

Keywords

Multi-label learning, Graph-based semi-supervised learning, Sylvester equation, Collaborative filtering

1 Introduction

Many learning problems require each instance to be assigned to multiple different categories, which are generally called multi-label learning problems. Multi-label learning problems arise in many practical applications such as automatic image annotation and text categorization. For example, in automatic image annotation, an image can be annotated as "road" as well as "car", where the terms "road" and "car" are different categories. Similarly, in text categorization, each document usually has different topics (e.g. "politics", "economy" and "military"), where different topics are different categories.

The most common approach toward multi-label learning is to decompose it into multiple independent binary classification problems, one for each category. The final labels for each instance can be determined by combining the classification results from all the binary classifiers. The advantage of this method is that many state-of-the-art binary classifiers can be readily used to build a multi-label learning machine. However, this approach ignores the underlying mutual correlations among different categories, while in practice, which usually do exist and could have significant contributions to the classification performance. Zhu *et al.* [35] gives an example illustrating the importance of considering the category correlations. To take the dependencies among categories into account, a straightforward approach is to transform the multi-label learning problem into a set of binary classification problems where each possible combination of categories rather than each category is regarded as a new class. In other words, a multi-label learning problem with n different categories would be converted into $2^n - 1$ binary classification problems where each class corresponds to a possible combination of the original categories. However, this approach has two serious drawbacks. First, when the number of original categories is quite large, the number of the combined classes, which increases exponentially, would become too large to be tractable; Second, when there are very few instances in many combined classes, the data sparsity problem would occur. So this approach is limited to a relatively small number of categories and assumes that the amount of training data is sufficient for training each binary classifier. In the past years, many novel multi-label learning algorithms modeling the correlations among categories have been developed [3, 7, 9–11, 14, 16, 17, 21, 22, 28–30, 33, 35], some of which will be introduced briefly in Section 2.

In this paper, we present a novel Semi-supervised Multi-label learning framework by solving a *Sylvester Equation* (*SMSE*). Two graphs are first constructed on

*State Key Laboratory on Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology(TNList), Tsinghua University, Beijing 100084, P. R. China, {g-c05, songyq99, feiwang03}@mails.thu.edu.cn, zcs@mail.thu.edu.cn

instance level and *category* level respectively. For *instance* level, a graph is defined based on both labeled and unlabeled instances, where each node represents one instance and each edge weight reflects the similarity between corresponding pairwise instances; Similarly, for *category* level, a graph is also built based on all the categories, where each node represents one category and each edge reflects the similarity between corresponding pairwise categories. Then we define a quadratic energy function on each graph, and by minimizing the combination of the two energy functions that balance the two energy terms, the labels of unlabeled data can be inferred. Here, the correlations among different categories have been considered via the energy function for *category* graph. In fact, our algorithm can be viewed as a regularization framework including two regularization terms corresponding to the two energy functions respectively. The regularization term for *instance* graph measures the smoothness of the labels of instances, and the regularization term for *category* graph measures the smoothness of the labels of categories. Finally, the labels of unlabeled instances can be obtained by solving a *Sylvester Equation*.

The rest of this paper is organized as follows: we first give a brief summary of related work on multi-label learning in Section 2; Section 3 describes our semi-supervised multi-label learning algorithm; in Section 4, we discuss our algorithm' relationship with spectral clustering; the data and experimental results are presented in Section 5; Section 6 presents our algorithm' extended application on collaborative filtering, followed by our conclusions in Section 7.

2 Related Work

Just as discussed in Section 1, the most simple method toward multi-label learning is to divide it into a set of binary classification problems, one for each category [6, 15, 32]. This approach suffers from a number of disadvantages. One disadvantage is that it can not scale to a large number of categories since a binary classifier has to be built for each category. Another disadvantage is that it does not exploit the correlations among different categories, because each category is treated independently. Finally, this approach may face the severe unbalanced data problem especially when the number of categories is large. When the number of categories is large, the number of "negative" instances for each binary classification problem could be quite larger than the number of "positive" instances. Consequently, the binary classifiers is likely to output the "negative" labels for most "positive" instances.

Another direction toward multi-label learning is label ranking [7–9, 25]. These approaches learn a ranking function of category labels from the labeled instances and apply it to classify each unknown test instance by choosing all the categories with the scores above the given threshold. Compared with the above binary classification approach, the label ranking approaches can be more appropriate to deal with large number of categories because only one ranking function need to be learned to compare the relevance of individual category labels with respect to test instances. The label ranking approaches also avoid the unbalanced data problem since they do not make binary decisions on category labels. Although the label ranking approaches provide a unique way to handle the multi-label learning problem, they do not exploit the correlations among data categories either.

Recently, more and more approaches for multi-label learning that consider the correlations among categories have been developed. Ueda *et al.* [30] suggests a generative model which incorporates the pairwise correlation between any two categories into multi-label learning. Griffiths *et al.* [12] proposes a Bayesian model to determine instance labels via underlying latent representations. Zhu *et al.* [35] employs a maximum entropy method for multi-label learning to model the correlations among categories. McCallum [22] and Yu *et al.* [33] apply approaches based on latent variables to capture the correlations among different categories. Cai *et al.* [5] and Rousu *et al.* [23] assume a hierarchical structure among the categories labels to handle the correlation information among categories. Kang *et al.* [16] gives a correlated label propagation framework for multi-label learning that explicitly exploits the correlations among categories. Unlike the previous work that only consider the correlations among different categories, Liu *et al.* [21] presents a semi-supervised multi-label learning method. It is based on constrained non-negative matrix factorization which exploits unlabeled data as well as category correlations. Generally, in comparision with supervised methods, semi-supervised methods can effectively make use of the information provided by unlabeled instances, and are superior particularly when the number of training data is relatively small. In this paper, we propose a novel semi-supervised approach for multi-label learning different from [21].

3 Semi-supervised Multi-label Learning by Solving a Sylvester Equation

We will first introduce some notations that will be used throughout the paper. Suppose there are l labeled instances $(x_1, y_1), \cdots, (x_l, y_l)$, and u unlabeled instances $x_{l+1}, \cdots, x_{l+u}$, where each $x_i = (x_{i1}, \cdots, x_{im})^T$ is an m-dimensional feature vector and each $y_i = (y_{i1}, \cdots, y_{ik})^T$ is a k-dimensional label vector. Here, we assume

the label of each instance for each category is binary: $y_{ij} \in \{0, 1\}$. Let $n = l + u$ be the total number of instances, $X = (x_1, \cdots, x_n)^T$ and $Y = (y_1, \cdots, y_n)^T = (c_1, \cdots, c_k)$.

3.1 Background Our work is related to semi-supervised learning, for which Seeger [26], Zhu [36] give a detailed description respectively. In order to make our work more comprehensive, we will introduce Zhu *et al.*'s graph-based semi-supervised learning algorithm [37].

Consider a connected graph $G = (V, E)$ with nodes corresponding to the n instances, where nodes $L = \{1, \cdots, l\}$ correspond to the labeled instances with labels $y_1, \cdots, y_l$, and nodes $U = \{l + 1, \cdots, l + u\}$ correspond to the unlabeled instances. The object is to predict the labels of nodes U. We define an $n \times n$ symmetric weight matrix W on the edges of the graph as follows

$$W_{ij} = \exp(-\sum_{d=1}^{m} \frac{(x_{id} - x_{jd})^2}{\sigma_d^2}) \tag{3.1}$$

where $\sigma_1, \cdots, \sigma_m$ are length scale hyperparameters for each dimension. Thus, the nearer the nodes are, the larger the corresponding edge weight is. For reducing parameter tuning work, we generally suppose $\sigma_1 = \cdots = \sigma_m$.

Define a real-valued function $f : V \to \mathbb{R}$ that determines the labels of unlabel instances. We constrain that f satisfies $f_i = y_i (i = 1, \cdots, l)$. Assume that nearby points on the graph are likely to have similar labels, which motivates the choice of the quadratic energy function

$$E(f) = \frac{1}{2} \sum_{i,j=1}^{n} W_{ij} \|f_i - f_j\|^2 \tag{3.2}$$

By minimizing the above energy function, the soft labels of unlabeled instances can be computed. Further, the optimization problem can be summarized as follows [37]

$$\min \infty \sum_{i=1}^{l} \|f_i - y_i\|^2 + \frac{1}{2} \sum_{i,j=1}^{n} W_{ij} \|f_i - f_j\|^2 \tag{3.3}$$

Essentially, here the energy function as a regularization term measures the smoothness of the labels of instances. Zhou *et al.* [34] gives a similar semi-supervised learning algorithm, which can be described as the following optimization problem

$$\min \quad \mu \sum_{i=1}^{n} \|f_i - y_i\|^2 + \frac{1}{2} \sum_{i,j=1}^{n} W_{ij} \|f_i / \sqrt{d_i} - f_j / \sqrt{d_j}\|^2 \tag{3.4}$$

where μ is a positive constant, $d_i = \sum_{j=1}^{n} W_{ij}$ and $y_i = 0 (i = l + 1, \cdots, n)$. Furthermore, Belkin *et al.* [2] proposes a unified regularization framework for semi-supervised learning by introducing an additional regularization term in *Reproducing Kernel Hilbert Space* (*RKHS*).

3.2 Our Basic Framework Traditional graph-based semi-supervised methods only construct a graph on *instance* level, which is appropriate when there are no correlations among categories. However, category correlations often exist in a typical multi-label learning scenario. Therefore, in order to make use of the correlation information, we have another graph constructed on *category* level too. Let $G' = (V', E')$ denote the *category* graph with k nodes, where each node represents one category. We define a $k \times k$ symmetric weight matrix W' as the following formula

$$W'_{ij} = \exp(-\lambda(1 - \cos(c_i, c_j))) \tag{3.5}$$

where λ is a hyperparameter, c_i is a binary vector whose elements are set to be one when the corresponding training instances belong to the ith category and zero otherwise (Please refer to the notation c_i at the beginning of Section 3) and $\cos(c_i, c_j)$ computes the *Cosine Similarity* between c_i and c_j by

$$\cos(c_i, c_j) = \frac{\langle c_i, c_j \rangle}{\|c_i\| \|c_j\|} \tag{3.6}$$

Define $F = (f_1, \cdots, f_n)^T = (g_1, \cdots, g_k)$, and we can also obtain a quadratic energy function for *category* graph

$$E'(g) = \frac{1}{2} \sum_{i,j=1}^{k} W'_{ij} \|g_i - g_j\|^2 \tag{3.7}$$

This can also been viewed as a regularization term that measures the smoothness of the labels of categories.

If we incorporate the regularization term for *category* graph into Eq. (3.3), the category correlation information can be used effectively. This encourages us to propose the following graph-based semi-supervised algorithm for multi-label learning, i.e. *SMSE1*

$$\min \infty \sum_{i=1}^{l} \|f_i - y_i\|^2 + \mu E(f) + \nu E'(g) \tag{3.8}$$

where μ and ν are nonnegative constants that balance $E(f)$ and $E'(g)$. By solving Eq. (3.8), we can

obtain the soft labels for unlabeled instances that in fact provide a ranked list of category labels for each unlabeled instance.

Similarly, if the regularization term for *category* graph is incorporated into Eq. (3.4), we can obtain another semi-supervised algorithm for multi-label learning, i.e. *SMSE2*

$$\begin{aligned} \min \quad & \sum_{i=1}^{l} \|f_i - y_i\|^2 + \\ & \frac{1}{2}\beta \sum_{i,j=1}^{n} W_{ij} \|f_i/\sqrt{d_i} - f_j/\sqrt{d_j}\|^2 + \\ & \frac{1}{2}\gamma \sum_{i,j=1}^{k} W'_{ij} \|g_i/\sqrt{d'_i} - g_j/\sqrt{d'_j}\|^2 \end{aligned} \tag{3.9}$$

where β and γ are nonnegative constants and $d'_i = \sum_{j=1}^{n} W'_{ij}$.

Next we will give the solution of Eq. (3.8) and Eq. (3.9).

3.3 Solving the *SMSE*

3.3.1 ***SMSE1*** First we have

$$\begin{aligned} E(f) &= \frac{1}{2}\sum_{i,j=1}^{n} W_{ij}\|f_i - f_j\|^2 \\ &= \frac{1}{2}\sum_{i,j=1}^{n} W_{ij}(f_i^T f_i + f_j^T f_j - 2f_i^T f_j) \\ &= \frac{1}{2}(\sum_{i=1}^{n} d_i f_i^T f_i + \sum_{j=1}^{n} d_j f_j^T f_j - 2\sum_{i,j=1}^{n} W_{ij} f_i^T f_j) \\ &= \mathrm{trace}(F^T(D-W)F) \\ &= \mathrm{trace}(F^T L F) \end{aligned} \tag{3.10}$$

where $d_i = \sum_{j=1}^{n} W_{ij}$, $D = \mathrm{diag}(d_i)$ and $L = D - W$. Here L is called the *combinatorial Laplacian*, and obviously is symmetric.

Similarly,

$$\begin{aligned} E'(g) &= \frac{1}{2}\sum_{i,j=1}^{k} W'_{ij}\|g_i - g_j\|^2 \\ &= \mathrm{trace}(F(D'-W')F^T) \\ &= \mathrm{trace}(FHF^T) \end{aligned} \tag{3.11}$$

where $D' = \mathrm{diag}(d'_i)$, $d'_i = \sum_{j=1}^{k} W'_{ij}$ and $H = D' - W'$. H is the *combinatorial Laplacian* of *category* graph.

Therefore, Eq. (3.8) reduces to finding

$$\begin{aligned} \min \quad & \mu\mathrm{trace}(F^T L F) + \nu\mathrm{trace}(FHF^T) \\ \text{s.t.} \quad & f_i = y_i (i = 1, \cdots, l) \end{aligned} \tag{3.12}$$

In order to solve the above optimization problem, let $\alpha = (\alpha_1, \cdots, \alpha_l)^T$ be the $l \times k$ *Lagrange* multiplier matrix for the constraint $f_i = y_i (i = 1, \cdots, l)$. The *Lagrange* function $\mathrm{Lag}(F, \alpha)$ becomes

$$\begin{aligned} \mathrm{Lag}(F,\alpha) &= \mu\mathrm{trace}(F^T L F) + \nu\mathrm{trace}(FHF^T) + \\ & \sum_{i=1}^{l} \alpha_i^T (f_i - y_i) \end{aligned} \tag{3.13}$$

By applying the matrix properties $\partial\mathrm{trace}(X^T A X)/\partial X = (A + A^T)X$ and $\partial\mathrm{trace}(XAX^T)/\partial X = X(A + A^T)$, the *Kuhn-Turker* condition $\partial\mathrm{Lag}(F,\alpha)/\partial F = 0$ becomes

$$\mu LF + \nu FH + \frac{1}{2}\begin{pmatrix} \alpha \\ 0 \end{pmatrix} = 0 \tag{3.14}$$

We split the matrix L into four blocks after the lth row and column: $L = \begin{pmatrix} L_{ll} & L_{lu} \\ L_{ul} & L_{uu} \end{pmatrix}$ and let $F = \begin{pmatrix} F_l \\ F_u \end{pmatrix}$ where F_u denotes the soft labels of unlabeled instances. So the following equation can be derived from Eq. (3.14)

$$\mu L_{ul} F_l + \mu L_{uu} F_u + \nu F_u H = 0 \tag{3.15}$$

The above matrix equation is called *Sylvester Equation* which often occurs in the control domain. We first discuss the solutions of *Sylvester Equation*

$$AX + XB = C \tag{3.16}$$

where $A \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{n\times n}$ and $X, C \in \mathbb{R}^{m\times n}$.

THEOREM 3.1. *Eq. (3.16) has a solution if and only if the matrices*

$$\begin{pmatrix} A & 0 \\ 0 & -B \end{pmatrix} \text{and} \begin{pmatrix} A & C \\ 0 & -B \end{pmatrix} \tag{3.17}$$

are similar.

THEOREM 3.2. *When Eq. (3.16) is solvable, it has a unique solution if and only if the eigenvalues $\delta_1, \cdots, \delta_u$ of A and $\gamma_1, \cdots, \gamma_k$ of B satisfy $\delta_i + \gamma_j \neq 0$ $(i = 1, \cdots, u; j = 1, \cdots, k)$.*

Please see [18] for the proofs of Thm. 3.1 and 3.2.

Eq. (3.15) has a unique solution F_u if it satisfies the above conditions that usually easily occur in the practical multi-label learning problems.

Here, an iterative Krylov-subspace method is adopted to solve *Sylvester Equation*. Please see [13] for details.

When $\nu = 0$ and $\mu \neq 0$, Eq. (3.15) becomes

$$L_{ul}F_l + L_{uu}F_u = 0 \tag{3.18}$$

This corresponds to solving the optimization problem in Eq. (3.3), so Zhu *et al.*'s semi-supervised learning approach [37] can be viewed as a special case of *SMSE1*.

3.3.2 *SMSE2* First

$$\begin{aligned}
&\frac{1}{2}\sum_{i,j=1}^{n} W_{ij}\|f_i/\sqrt{d_i} - f_j/\sqrt{d_j}\|^2 \\
=\ &\frac{1}{2}\sum_{i,j=1}^{n} W_{ij}(f_i^T f_i/d_i + f_j^T f_j/d_j - 2f_i^T f_j/\sqrt{d_i d_j}) \\
=\ &\frac{1}{2}(\sum_{i=1}^{n} f_i^T f_i + \sum_{j=1}^{n} f_j^T f_j - 2\sum_{i,j=1}^{n} W_{ij} f_i^T f_j/\sqrt{d_i d_j}) \\
=\ &\mathrm{trace}(F^T(I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}})F) \\
=\ &\mathrm{trace}(F^T L_n F)
\end{aligned} \tag{3.19}$$

where $L_n = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$. Here, L_n is called the *normalized Laplacian* and also is symmetric.

Similarly,

$$\begin{aligned}
&\frac{1}{2}\sum_{i,j=1}^{k} W'_{ij}\|g_i/\sqrt{d'_i} - g_j/\sqrt{d'_j}\|^2 \\
=\ &\mathrm{trace}(F(I - D'^{-\frac{1}{2}}W'D'^{-\frac{1}{2}})F^T) \\
=\ &\mathrm{trace}(FH_nF^T)
\end{aligned} \tag{3.20}$$

where $H_n = I - D'^{-\frac{1}{2}}W'D'^{-\frac{1}{2}}$. H_n is the *normalized Laplacian* of *category* graph.

Thus, Eq. (3.9) is converted into

$$\min \sum_{i=1}^{n} \|f_i - y_i\|^2 + \beta\mathrm{trace}(F^T L_n F) + \gamma\mathrm{trace}(FH_nF^T) \tag{3.21}$$

By applying the optimization method similar to solving *SMSE1*, Eq. (3.21) reduces to

$$(\beta L_n + I)F + \gamma FH_n - Y = 0 \tag{3.22}$$

Obviously, the above matrix equation is also a *Sylvester Equation*. Compared with solving *SMSE1*, solving *SMSE2* do not refer to block matrices but need more computational expense since the number of variables increases from $u \times k$ to $n \times k$. However, *SMSE1* cannot be applied in some cases where no natural block matrix exists while *SMSE2* can. In Section 6, we will give such an application.

4 Connections to Spectral Clustering

Zhu *et al.* [37] discussed the relations between their graph-based semi-supervised learning algorithm and spectral clustering. Spectral clustering is unsupervised where there is no labeled information and only depends on the graph weights W. On the other hand, Graph-based semi-supervised learning algorithms maintain a balance between how good the clustering is, and how well the labeled data can be explained by it [36].

The typical spectral clustering approach: the normalized cut [27] seeks to minimize

$$\begin{aligned}
\min\quad & \frac{y^T(D-W)y}{y^T D y} \\
\text{s.t.}\quad & y^T D\mathbf{1} = 0
\end{aligned} \tag{4.23}$$

The solution y is the second smallest eigenvector of the generalized eigenvalue problem $Ly = \lambda Dy$. Then y is discretized to obtain the clusters. In fact, if we add the labeled data information into Eq. (4.23) and simultaneously discard the scale constraint term $y^T Dy$, Zhu *et al.*'s semi-supervised learning algorithm [37] can be immediately obtained.

Therefore, if there is not any supervised labeled information and the graph weights W, W' both *instance* and *category* can be calculated in some way, our algorithm *SMSE* can reduce to do simultaneously clustering (also called co-clustering) on two different graphs. For example, if we apply the *combinational Laplacian* to do co-clustering, the corresponding co-clustering algorithm can be formalized as follows

$$\begin{aligned}
\min\quad & \frac{\mathrm{trace}(F^T LF)}{\mathrm{trace}(F^T DF)} + \tau\frac{\mathrm{trace}(FHF^T)}{\mathrm{trace}(FD'F^T)} \\
\text{s.t.}\quad & f_i^T D\mathbf{1} = 0,\ g_i^T D'\mathbf{1} = 0
\end{aligned} \tag{4.24}$$

where τ is a nonnegative constant. The solution F of the above equation is further done row and column clustering respectively. Thus, the clusters for both *category* and *instance* can be gotten. However, research on co-clustering has gone beyond the scope of this paper, and here we only concentrate on semi-supervised learning.

5 Experiments

5.1 Data Set and Experimental Setup Our data set is a subset of *RCV1-v2* text data, provided by

Reuters and corrected by Lewis et al. [19]. The data set includes the information of topics, regions and industries for each document and a hierarchical structure for topics and industries. Here, we use topics as the classification tasks and simply ignore the topic hierarchy. We first randomly pick 3000 documents, then choose words with more than 5 occurrences and topics with more than 40 positive assignments. Finally, We have 3000 documents with 4082 words, and have 60 topics left. On average, each topic contains 225 positive documents, and each document is assigned to 4.5 categories.

In order to reduce computational expense, we create kNN graphs rather than fully connected graphs. It means that nodes i and j are connected by an edge if i is in j's k-nearest-neighborhood or vice versa. Computation on such sparse graphs are fast. In general, the size of neighbors and other parameters in Eq. (3.8) and Eq. (3.9) can be gotten by doing cross validation on training set. In the next experiments, the sizes of neighbors for *instance* graph and *category* graph are 17 and 8 respectively.

5.2 Evaluation Metrics Since our approach only produces a ranked list of category labels for a test instance, in this paper we focus on evaluating the quality of category ranking. More concretely, we evaluate the performance when varying the number of predicted labels for each test instance along the ranked list of class labels. Following [16, 32], we choose F_1 *Micro* measure as the evaluation metric, which can be seen as the weighted average of $F1$ scores over all the categories (see [32] for details) . The F_1 measure of the sth category is defined as follows

$$F_1(s) = \frac{2p_s r_s}{p_s + r_s} \tag{5.25}$$

where p_s and r_s are the precision and recall of the sth category, respectively. And they can be calculated by using the following equations

$$p_s = \frac{|\{x_i | s \in C_i \bigwedge s \in \widehat{C}_i\}|}{|\{x_i | s \in \widehat{C}_i\}|} \tag{5.26}$$

$$r_s = \frac{|\{x_i | s \in C_i \bigwedge s \in \widehat{C}_i\}|}{|\{x_i | s \in C_i\}|} \tag{5.27}$$

where C_i and $\widehat{C}_i$ are the ith instance x_i's true labels and predicted labels, respectively.

5.3 The Influence of Parameters We analyze the influence of parameters in *SMSE*. We randomly choose 500 from 3000 documents as labeled data, and the left 2500 documents as unlabeled data. The number of predicted labels for each test document is assigned to 10.

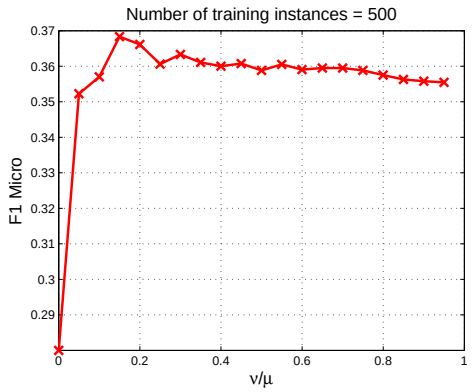


Figure 1: Performance of *SMSE1* with respect to ν/μ

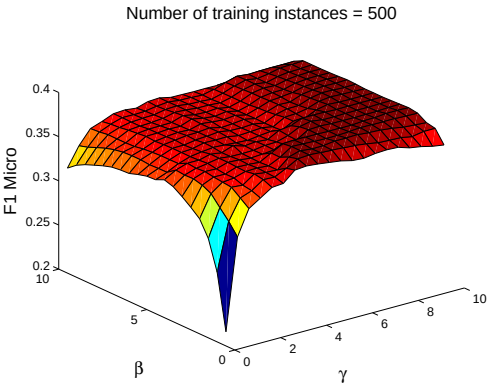


Figure 2: Performance of *SMSE2* with respect to β and γ

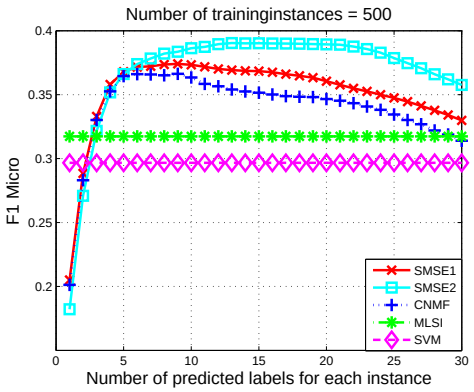
Set the hyperparameters $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_m^2 = 0.27$, $\lambda = 10$. With respect to above configurations, Fig. 1 shows the $F1$ *Micro* scores of *SMSE1* when varying the value of ν/μ. When $\nu = 0$ and $\mu \neq 0$, *SMSE1* reduces to Eq. (3.3) that only constructs an *instance* graph and does not consider the correlations among different topics. Contrarily, when $\nu \neq 0$ and $\mu = 0$, only a *category* graph is used in *SMSE1*. From Fig. 1, we see that by choosing the appropriate value of ν/μ our approach indeed makes use of the correlation information among different categories and obviously increases the performance compared with the algorithm which only builds an *instance* graph or a *category* graph. In practice, we only have a parameter tuned when fixing the other parameter to 1 in *SMSE1*.

Fig. 2 shows $F1$ *Micro* scores of *SMSE2* when varying the values of β and γ. Similarly, by choosing appropriate values of the two parameters, we can achieve the best predictions. However, in comparision with *SMSE1*, *SMSE2* has two parameters tuned rather than one.

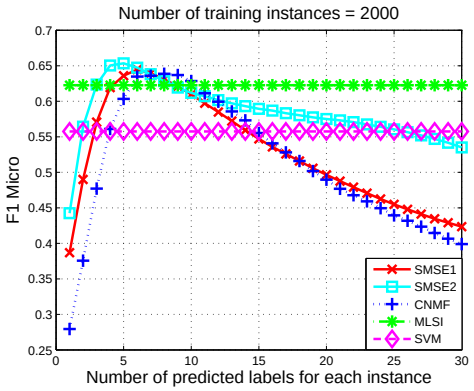
5.4 Comparisons and Discussions We compare our algorithm with three baseline models. The first one is a semi-supervised multi-label learning method based on *Constrained Non-negative Matrix Factorization* (*CNMF*) [21]. The key assumption behind *CNMF* is that two instances tend to have large overlap in their assigned category memberships if they share high similarity in their input patterns. *CNMF* evaluates the instance similarity matrix for *instance* graph from two different viewpoints: one is based on the correlations between the input patterns of these two instances, the other is based on the overlap between the category labels of these two instances. By minimizing the difference of the two similarity matrix, *CNMF* can determine the labels of unlabeled data. The second model is *Support Vector Machine* (*SVM*). A linear *SVM* classifier is built for each category independently. The last baseline model is *Multi-label Informed Latent Semantic Indexing* (*MLSI*) [33], which first maps the input features into a new feature space that retains the information of original inputs and meanwhile captures the dependency of the output labels, then trains a set of linear *SVM*s on this projected space. Fig. 3 shows the performance of all the five algoritms: *SMSE1*, *SMSE2*, *CNMF*, *SVM*, *MLSI* at different ranks when the number of training data is 500 or 2000. All the methods are tested by a 10-fold experiment using the same training/test split of the data set and the average of $F1$ *Micro* scores for each method is computed. It should be also noted that all parameters contained in the five methods are chosen by grid search. From Fig. 3, we can obtain:

1. *SMSE1*, *SMSE2* and *CNMF* achieve the similar performance in $F1$ *Micro*, and they both are superior to *SVM* and *MLSI* if we choose a proper number of predicted labels for each test instance. However, in comparision with *SMSE1* and *SMSE2*, *CNMF* have more variables and more complicated formulae to be calculated. The average execution time for *SMSE1*, *SMSE2* and *CNMF* on the *PC* with $2.4GHz$ *CPU* and $1Gb$ *RAM* using *Matlab* codes is 83.2s, 187.3s, and 423.9s respectively when the number of labeled data is 500, and 40.1s, 199.4s and 353.3s respectively when the number of labeled data is 2000. This sufficiently demonstrates *SMSE*'s advantage on computational expense. Just as discussed in Section 3.3.2, *SMSE2* has more variables to solve than *SMSE1* so that its execution time is more than that of *SMSE1*.

2. More performance improvement by *SMSE1*, *SMSE2* and *CNMF* is observed when the number of training data is 500 than when the number of training data is 2000. This is because that in semi-supervised learning the benefit provided by unlabeled data is expected to decrease with more labeled data, which has been verified in many studies such as [26, 36, 37].



(a)



(b)

Figure 3: Performance when varying the number of predicted labels for each test instance along the ranked list of category labels

To sum up, when the amount of labeled data is relatively small especially for labeled instances are difficult, expensive, or time consuming to obtain, relative to supervised algorithms, semi-supervised algorithms is generally a better choice. Here, considering the balance between $F1$ *Micro* and computational efficiency, maybe the overall performance of *SMSE1* is the best in the five approaches for multi-label learning.

6 The Extended Application on Collaborative Filtering

6.1 Introduction to Collaborative Filtering Collaborative filtering aims at predicting a test user's ratings for new items based on a collection of other like-minded user' ratings information. The key assumption

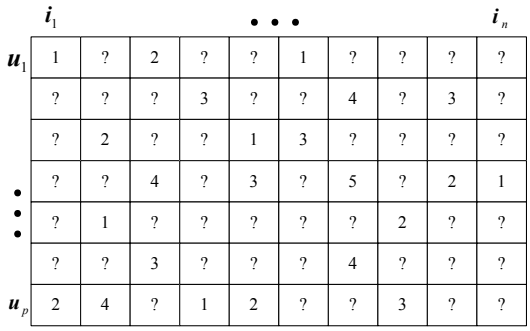


Figure 4: A user-item matrix. "?" means the item is not rated by the corresponding user.

is that users sharing the same ratings on past items tend to agree on new items. Various collaborative filtering techniques have been successfully utilized to build recommender systems (e.g. movies [1] and books [20]).

In a typical collaborative filtering scenario, there is a $p \times n$ user-item matrix $\boldsymbol{X}$ (see Fig. 4), where p is the number of users and n is the number of items. Each element of $\boldsymbol{X}$: $x_{jm} = r$ denotes that the jth user rates the mth item by r, where $r \in \{1, \cdots, R\}$. When the item is not rated, $x_{jm} = \emptyset$. The goal is usually to predict the unknown items' ratings.

Let

$$\boldsymbol{X} = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_p]^T, \ \boldsymbol{u}_j = (x_{j1}, \cdots, x_{jn})^T, \quad j \in \{1, \cdots, p\} \tag{6.28}$$

where the vector $\boldsymbol{u}_j$ indicates the jth user's ratings to all items.

Likewise, the user-item matrix $\boldsymbol{X}$ can be decomposed into column vectors

$$\boldsymbol{X} = [\boldsymbol{i}_1, \cdots, \boldsymbol{i}_n], \ \boldsymbol{i}_m = (x_{1m}, \cdots, x_{pm})^T, \quad m \in \{1, \cdots, n\} \tag{6.29}$$

where the vector $\boldsymbol{i}_m$ indicates all users' ratings to the mth item.

Collaborative filtering approaches can be mainly divided into two categories: user-based [4] and item-based [24]. User-based algorithms for collaborative filtering aim at predicting a test user's ratings for unknown items by synthesizing the like-minded users' information. It first computes the similarities between the test user and other users, then selects the most similar K-users to the test user by ranking the similarities. Finally the unknown rating is predicted by combining the known rating of the K neighbors. Item-based algorithms for collaborative filtering are similar to user-based algorithms except for that they need to compute the pairwise similarity between the items. In item-based approaches, the similarities between the test item and other items are also first calculated and sorted, as a result, we can obtain the most similar K-items to the test item. Then, the unknown rating is also predicted by combining the known rating of the K neighbors.

6.2 Applying *SMSE2* on Collaborative filtering In fact, collaborative filtering is quite analogous to multi-label learning. If considering collaborative filtering from the viewpoint of graph, we can construct a user graph and an item graph respectively. The graph weights can be obtained by computing the similarities between pairwise user or item vectors (Here, we utilized Eq. (3.5) to calculate graph weights). The regularization term for user graph measures the smoothness of user vectors and the regularization term for item graph measures the smoothness of item vectors. Obviously, by combining the two regularization terms for user and item graphs, the unknown ratings can be gotten by solving the *SMSE*. It should be noted that since the user-item matrix does not have natural blocks (see Fig. 4), only *SMSE2* can be used in collaborative filtering while *SMSE1* cannot. To some extent, *SMSE2* can be seen as a hybrid method for collaborative filtering that convexly combines user-based and item-based.

6.3 Preliminary Experiments We used the *MovieLens* [1] data set to evaluate our algorithm. The *MovieLens* data set is composed of 943 users and 1682 items (1-5 scales), where each user has more than 20 ratings. Here, we extracted a subset which contained 500 users with more than 40 ratings and 1000 items. The first 300 users in the data set are selected into training set and the left 200 users as test set. In our experiments, the available ratings of each test user are half-and-half split into an observed set and a held out set. The observed ratings are used to predict the held out ratings.

Here, we are only concerned with ranking the unrated data and recommending the top ones to the active user. Therefore, following [31], we choose *Order Consistency* (*OC*) to measure how similar the predicted order to the true order. Assuming there are n items, v is the vector that these n items are sorted in an decreasing order according to their predicted ranking scores, v' is the vector that these n items are sorted in an decreasing order according to their true ratings. For these n items, we have $C_n^2 = n(n-1)/2$ ways to randomly select a pair of different items. $\mathcal{A}$ is the set whose elements are pairwise items whose relative order in v are the same as

[1] http://www. grouplens. org/

Table 1: The *OC* values of *SMSE2, IRSM and URSM*. A larger value means a better performance

Algorithm	*SMSE2*	*IRSM*	*URSM*	*IB*	*UB*
OC	**0.820**	0.785	0.782	0.719	0.711

in v', then *Order Consistency* is defined as

$$OC = |\mathcal{A}|/C_n^2 \tag{6.30}$$

The larger the value of *OC*, the better the predictions are.

Recently, Wang *et al.* [31] proposed a novel item-based recommendation scheme called *Item Rating Smoothness Maximization* (*IRSM*). In their framework, the items are first described by an undirected weighted graph, then based on Zhou *et al.*'s method [34], the unknown ratings can be predicted. Their theoretical analysis and experimental results show the effectiveness of *IRSM* on recommendation problems. It is easy to find that *IRSM* is a special case of *SMSE2*. In *IRSM*, the user graph is not been utilized. Similarly, if we only consider constructing a user graph to predict the unknown ratings, the other method that we call *User Rating Smoothness Maximization* (*URSM*) is obtained. It is clear that *URSM* is also a special case of *SMSE2*. Here, we compare *SMSE2* with four approaches including *IRSM*, *URSM*, traditional user-based (*UB*) [4] and item-based (*IB*) [24]. Tab. 1 shows the *OC* values of the five algorithms. Note that all parameters are determined by grid search. It can be observed that *SMSE2* is superior to other four approaches, that validates the effectiveness of *SMSE2* on collaborative filtering.

7 Conclusions

In this paper we propose a novel semi-supervised algorithm for multi-label learning by solving a *Sylvester Equation*. Two graphs are first constructed on both *instance* level and *category* level respectively. By combining the regularization terms for the two graphs, a regularization framework for multi-label learning is suggested. The labels of unlabeled instances can be obtained by solving a *Sylvester Equation*. Our method can exploit unlabeled data information and the correlations among categories. Empirical studies show that our algorithm is quite competitive against state-of-the-art multi-label learning techniques. Additionally, we successfully applied our algorithm to collaborative filtering.

In the future, we will further study *SMSE2*'s overall performance on collaborative filtering and develop more effective multi-label learning approaches.

References

[1] http://movielens.umn.edu.

[2] M. Belkin, P. Niyogi, and V. Sindhwani. On manifold regularization. In *Proc. of AISTATS*, 2005.

[3] M. R. Boutella, X. Luoh, J.and Shen, and C. M. Browna. Learning multi-label scene classification. *Pattern Recognition*, 37:1757–1771, 2004.

[4] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. of UAI*, 1998.

[5] L. Cai and T. Hofmann. Hierarchical document categorization with support vector machines. In *Proc. of CIKM*, 2004.

[6] E. Chang, K. Goh, G. Sychay, and G. Wu. Content-based soft annotation for multimodal image retrieval using bayes point machines. *IEEE Tran. on Circuits and Systems for Video Tech. Special Issue on Conceptual and Dynamical Aspects of Multimedia Content Description*, 13(1), 2003.

[7] K. Crammer and Y. Singer. A new family of online algorithms for category ranking. In *Proc. of SIGIR*, 2002.

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

[9] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *Proc. of NIPS*, 2001.

[10] S. Gao, W. Wu, C. H. Lee, and T. S. Chua. A mfom learning approach to robust multiclass multi-label text categorization. In *Proc. of ICML*, 2004.

[11] N. Ghamrawi and A. McCallum. Collective multi-label classification. In *Proc. of CIKM*, 2005.

[12] T. Griffiths and Z. Ghahramani. Infinite latent feature models and the indian buffet process. In *Proc. of NIPS*, 2005.

[13] D. Y. Hu and L. Reichel. Krylov-subspace methods for the sylvester equation. *Linear Algebra and Its Applications*, (172):283–313, 1992.

[14] R. Jin and Z. Ghahramani. Learning with multiple labels. In *Proc. of NIPS*, 2003.

[15] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proc. of ECML*, 1998.

[16] F. Kang, R. Jin, and R. Sukthankar. Correlated label propagation with application to multi-label learning. In *Proc. of CVPR*, 2006.

[17] H. Kazawa, T. Izumitani, H. Taira, and E. Maeda. Maximal margin labeling for multi-topic text categorization. In *Proc. of NIPS*, 2005.

[18] P. Lancaster and M. Tismenetsky. *The theory of matrices: with applications*. Academic Press, 1985.

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

[20] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filter-

ing. *IEEE Internet Cmoputing*, pages 76–80, January-February 2003.

[21] Y. Liu, R. Jin, and L. Yang. Semi-supervised multi-label learning by constrained non-negative matrix factorization. In *Proc. of AAAI*, 2006.

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

[23] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor. On maximum margin hierarchical multi-label classification. In *Proc. of NIPS Workshop on Learning With Structured Outputs*, 2004.

[24] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proc. of WWW*, 2001.

[25] R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2-3), 2000.

[26] M. Seeger. Learning with labeled and unlabeled data. Techinial report, University of Edinburgh, 2001.

[27] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.

[28] B. Tasker, V. Chatalbashev, and D. Koller. Learning associative markov networks. In *Proc. of ICML*, 2004.

[29] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proc. of ICML*, 2004.

[30] N. Ueda and K. Saito. Parametric metric models for multi-labelled text. In *Proc. of NIPS*, 2002.

[31] F. Wang, S. Ma, L. Yang, and T. Li. Recommendation on item graphs. In *Proc. of ICDM*, 2006.

[32] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1/2), 1999.

[33] K. Yu, S. Yu, and V. Tresp. Multi-label informed latent semantic indexing. In *Proc. of SIGIR*, 2005.

[34] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf. Learning with local and global consistency. In *Proc. of NIPS*, 2003.

[35] S. Zhu, X. Ji, W. Xu, and Y. Gong. Multi-labelled classification using maximum entropy method. In *Proc. of SIGIR*, 2005.

[36] X. Zhu. Semi-supervised learning literature survey. Techinial Report TR 1530, University of Wisconsin-Madision, 2006.

[37] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian random fields and harmonic functions. In *Proc. of ICML*, 2003.