JMLR: Workshop and Conference Proceedings 29:1–16, 2013 ACML 2013

Multi-Label Classification with Unlabeled Data: An Inductive Approach

Lei Wu WUL@SEU.EDU.CN and **Min-Ling Zhang** ZHANGML@SEU.EDU.CN
School of Computer Science and Engineering, MOE Key Laboratory of Computer Network and Information Integration, Southeast University, Nanjing 210096, China

Abstract

The problem of multi-label classification has attracted great interests in the last decade. Multi-label classification refers to the problems where an example that is represented by a *single* instance can be assigned to *more than one* category. Until now, most of the researches on multi-label classification have focused on supervised settings whose assumption is that large amount of labeled training data is available. Unfortunately, labeling training example is expensive and time-consuming, especially when it has more than one label. However, in many cases abundant unlabeled data is easy to obtain. Current attempts toward exploiting unlabeled data for multi-label classification work under the *transductive* setting, which aim at making predictions on existing unlabeled data while can not generalize to new unseen data. In this paper, the problem of *inductive* semi-supervised multi-label classification is studied, where a new approach named *iMLCU*, i.e. *inductive Multi-Label Classification with Unlabeled data*, is proposed. We formulate the inductive semi-supervised multi-label learning as an optimization problem of learning linear models and ConCave Convex Procedure *(CCCP)* is applied to optimize the non-convex optimization problem. Empirical studies on twelve diversified real-word multi-label learning tasks clearly validate the superiority of *iMLCU* against the other well-established multi-label learning approaches.

Keywords: multi-label learning, semi-supervised learning, unlabeled data

1. Introduction

Traditional supervised learning is one of the mostly-studied machine learning paradigms, where each real-word object (example) is represented by a single instance (feature vector) and associated with a single label which characterizes its semantics. However, many real-word objects might be complicated and have multiple semantic meanings, which make the above traditional supervised learning assumption not fit. For example, in automatic image annotation, an image can convey various messages, such as *boat, sea, sky* and *beach*; In text categorization, an article may include multiple topics, such as *politics, economics, parliamentary elections* and *unemployment rate*. In contrast to traditional supervised learning, in multi-label learning an object is also represented by a single instance while associated with a set of labels instead of a single label. The task is to learn a function which can predict proper label sets for unseen instances (Zhang and Zhou (in press)). Traditional two-class and multi-class learning can both be cast as special cases of multi-label learning problem where the size of an object's label set is one.

Conventional multi-label approaches focus on the supervised settings and have achieved much success. However, successful supervised learning requires sufficient amount of labeled

training examples. In many applications, labeling training example is extremely expensive and time-consuming, especially when it has more than one label. However, abundant unlabeled data is easy to obtain. Naturally, it is much desired that the large amount of unlabeled data can be utilized together with the limited amount of labeled data to improve the classification performance. Semi-supervised learning (Zhu and Goldberg (2009)) is one of the most popular strategies to achieve this goal, where unlabeled data is exploited to facilitate the learning process in addition to labeled data without human intervention.

Recently, several attempts have been made toward designing semi-supervised multi-label learning approaches (Zha et al. (2009)) (Kong et al. (2013)) (Chen et al. (2008)) (Wang et al. (2011)) (Guo and Schuurmans (2012)). All of these algorithms work under *transductive* setting, which aim at making predictions on existing unlabeled data while can not generalize to new unseen data. But in many real world applications, the requirement that all unlabeled data are available during training may not be satisfied. For example, in automatic image annotation, the image that we need to annotate may be unseen when we are inducing the annotation system. To adapt to this situation, we propose a new algorithm called *iMLCU* i.e. *inductive Multi-Label Classification with Unlabeled data* in this paper. We first formulate the inductive semi-supervised multi-label learning as an optimization problem of learning linear models, which fits labeled data by exploiting correlations among class labels and utilizes unlabeled data via appropriate regularizations. After that, the resulting optimization which is non-convex is solved via the ConCave Convex Procedure (*CCCP*). The effectiveness of *iMLCU* is thoroughly validated with comparative studies over a total of twelve benchmark multi-label data sets.

The rest of this paper is organized as follows. We give a brief summary of related work on semi-supervised multi-label classification in Section 2; Section 3 describes our inductive semi-supervised multi-label classification algorithm; The experimental data, setup as well as results are presented in Section 4; Finally, conclusion of our work is given in Section 5.

2. Related Work

In this section, we focus on reviewing closely related works on semi-supervised multi-label learning. For more information on multi-label learning in the general sense, the readers may refer to survey papers such as (Zhang and Zhou (in press)) and (Tsoumakas et al. (2010)).

Traditional supervised learning requires sufficient amount of labeled training examples which may not be easy to obtain in many real world applications. We usually need to handle the situation where a small size of labeled data with a large amount of unlabeled data are available. Under this condition, some semi-supervised multi-label algorithms are proposed. (Zha et al. (2009)) proposes a graph-based learning framework which employs two types of regularizer. One is used to prefer the label consistency on the graph and the other is adopted to prefer the correlations of multiple labels. (Kong et al. (2013)) formulates the transductive multi-label classification as an optimization problem of estimating label concept compositions and derives a closed-form solution to this optimization problem. In addition, the same idea is utilized to learn the *cardinality* of the label set for each unlabeled instance so that we can assign label sets to the unlabeled instances based upon the estimated label concept compositions. In (Chen et al. (2008)), a regularization framework combining two regularization terms for the two graphs, i.e. *instance graph* and *label graph*,

is suggested. (Wang et al. (2011)) presents an effective multi-label classification method that simultaneously models the labeling consistency between the visually similar videos and the multi-label interdependence for each video. (Guo and Schuurmans (2012)) proposes an algorithm that learns a subspace representation of the labeled and unlabeled data while simultaneously trains a supervised large-margin multi-label classifier on the labeled data.

Except (Guo and Schuurmans (2012)), the common strategy adopted by the aforementioned approaches is that they all construct the graph by utilizing labeled and unlabeled training examples as the vertices. As a major family of semi-supervised learning, graph-based methods have attracted significant interests due to their effectiveness and efficiency (Zhou et al. (2004))(Zhu et al. (2003)). Almost all graph-based methods essentially estimate a function on the graph such that it has two properties: 1) it should be close to the given labels on the labeled examples, and 2) it should be smooth on the whole graph. Graph-based methods differ slightly in the function they formulate on the graph. Due to the characteristics of graph construction, all the unlabeled examples must be available during training, i.e. all these existing semi-supervised multi-label classification methods are of transductive setting (Zhu and Goldberg (2009)) and the learned classifier can only work on the label set prediction of unlabeled data used during training while can not generalize to the new unseen data. For (Guo and Schuurmans (2012)), the subspace representation is induced from existing labeled and unlabeled data, which also works under transductive setting.

In this paper, the problem of *inductive* semi-supervised multi-label learning is studied, where the corresponding *iMLCU* approach is presented in the following section.

3. Our Approach

3.1. Problem Formulation

In this part, we will introduce some notations that will be used throughout the paper. Let $\mathcal{X} = \mathbb{R}^d$ be the d-dimensional feature space, and $\mathcal{Y} = \{y_1, y_2, \ldots, y_q\}$ be the label space with q possible class labels. Here we assume that each class label is binary: $y_i \in \{+1, -1\}, 1 \leq i \leq q$. Suppose there are l labeled instances and u unlabeled instances. So we can symbolize training set as $D = \{(\mathbf{x}_1, Y_1), \ldots, (\mathbf{x}_l, Y_l), \mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u}\}$, where each $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ is a d-dimensional feature vector and each $Y_i \subseteq \mathcal{Y}$ is the label set of $\mathbf{x}_i$. We denote the labeled instances and unlabeled instances in D as D_l and D_u respectively, i.e. $D_l = \{(\mathbf{x}_1, Y_1), \ldots, (\mathbf{x}_l, Y_l)\}$ and $D_u = \{\mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+u}\}$. The learning problem we are interested in is to find from the training set D a family of q real-value functions $f_i : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, where $f_i(\mathbf{x}, y_i)$ can be regarded as the *confidence* of $y_i \in \mathcal{Y}$ being a proper label of $\mathbf{x}$.

3.2. Algorithm Detail

Let the classifier model be composed of q linear classifiers $\mathbf{W} = \{(\mathbf{w}_j, b_j) | 1 \leq j \leq q\}$, where $\mathbf{w}_j \in \mathbb{R}^d$ and $b_j \in \mathbb{R}$ are the *weight vector* and *bias* for the j-th class label y_j. In our

approach, the following scheme to predict the label sets for test instances is adopted:

$$\begin{aligned}\widehat{Y} &= (\widehat{y}_1, \ldots, \widehat{y}_q) \\ &= sign(f_1(\mathbf{x}, y_1), \ldots, f_q(\mathbf{x}, y_q)) \\ &= sign(\langle \mathbf{w}_1, \mathbf{x} \rangle + b_1, \ldots, \langle \mathbf{w}_q, \mathbf{x} \rangle + b_q)\end{aligned} \tag{1}$$

where function $f_i(\mathbf{x}, y_i)$ is defined in Section 3.1 and formulated as $f_i(\mathbf{x}, y_i) = \langle \mathbf{w}_i, \mathbf{x} \rangle + b_i \quad (1 \leq i \leq q)$.

Generally speaking, two key issues have to be addressed in designing inductive-style semi-supervised multi-label learning algorithm. The first one is how to properly exploit label correlations in algorithmic design, which is deemed to be essential for learning from multi-label data successfully (Zhang and Zhou (in press)). Based on the order of correlations being considered, existing label correlation exploitation strategies can be categorized as *first-order*, *second-order*, and *high-order* ones. Specifically, second-order strategy tackles multi-label learning problem by considering *pairs* relations between labels, such as the ranking between relevant label and irrelevant label (Elisseeff and Weston (2002)) (Fürnkranz et al. (2008)), or interaction between any pair of labels (Zhu et al. (2005)) (Ghamrawi and McCallum (2005)). Compared to first-order strategy which totally ignores label correlations, second-order approach does exploit label correlations to some extent. On the other hand, compared to high-order strategy, second-order strategy usually leads to lower model and computational complexity.

Therefore, in this paper, *iMLCU* employs second-order strategy for label correlations modeling. Specifically, by considering classifier model's ranking ability on the labeled example's relevant-irrelevant labels, the decision boundaries for labeled example $(\mathbf{x}_i, Y_i)$ can be defined by the hyperplanes whose equations are $\langle \mathbf{w}_k - \mathbf{w}_l, \mathbf{x}_i \rangle + b_k - b_l = 0$, where $(y_k, y_l) \in Y_i \times \overline{Y_i}$ and $\langle a, b \rangle$ is the inner product of two vectors, i.e. $a^T b$. Accordingly, we make use of labeled data in D_l via maximum margin assumption, which leads to the following objective function (Elisseeff and Weston (2002)):

$$\min_{\mathbf{W}, \Xi} \sum_{k=1}^{q} \|\mathbf{w}_k\|^2 + C \sum_{i=1}^{l} \frac{1}{|Y_i||\overline{Y_i}|} \sum_{(y_k, y_l) \in Y_i \times \overline{Y_i}} \xi_{ikl}. \tag{2}$$

$$\begin{aligned}\text{subject to: } & \langle \mathbf{w}_k - \mathbf{w}_l, \mathbf{x}_i \rangle + b_k - b_l \geq 1 - \xi_{ikl} \\ & \xi_{ikl} \geq 0 \qquad (1 \leq i \leq l, (y_k, y_l) \in Y_i \times \overline{Y_i})\end{aligned}$$

Here, the first term in the objective function controls the model complexity, and the second term controls the empirical ranking loss over the labeled data. In addition, $\Xi = \{\xi_{ikl} | 1 \leq i \leq l, (y_k, y_l) \in Y_i \times \overline{Y_i}\}$ correspond to the slack variables and C is the tradeoff parameter between model complexity and empirical loss.

The second issue to be addressed is how to utilize unlabeled data in the learning process whose labels are unknown. For unlabeled instances, naturally we want to place them outside the margin and penalize the loss where some unlabeled instances lie within the margin or even on the wrong side of the decision boundary. But without knowing the labels of an unlabeled instance, we do not even know whether this unlabeled instance is on the correct or the wrong side of the decision boundary. In inspiration of (Joachims (1999)), we adapt

Table 1: Pseudo-codes of *iMLCU*.

$Y = \text{iMLCU}(D, C_1, C_2, \mathbf{u}, maxIter)$
Inputs:
D: the multi-label training set defined in Section 3.1
C_1 and C_2: the nonnegative balance papameter
$\mathbf{u}$: the unseen instance ($\mathbf{u} \in \mathcal{X}$)
$maxIter$: maximal number of iterations
Outputs:
Y: the predicted label set for $\mathbf{u}$ ($Y \subseteq \mathcal{Y}$)
Process:
Initiate $\mathbf{w}_v^0$ and b_v^0 from the labeled data $\quad (1 \le v \le q)$
repeat:
$iter \leftarrow 1$
for $v \leftarrow 1$ to q
$\widehat{y}_{jv} \leftarrow sign(\langle \mathbf{w}_v^{iter\text{-}1}, \mathbf{x}_j \rangle + b_v^{iter\text{-}1}) \quad (l+1 \le j \le l+u)$
learn $\mathbf{w}_v^{iter}$ and b_v^{iter} by optimizing Eq.(7)
endfor
$iter \leftarrow iter + 1$
until convergence of Eq.(7) or $iter$ exceeds $maxIter$
$Y \leftarrow sign(\langle \mathbf{w}_1, \mathbf{u} \rangle + b_1, \ldots, \langle \mathbf{w}_q, \mathbf{u} \rangle + b_q)$ according to Eq.(1)

the idea of *S3VM* to the multi-label data. We treat the prediction obtained from Eq.(1) as the *putative label sets* of unlabeled instance $\mathbf{x}$ and then penalize the loss on i-th label y_i by applying the hinge loss function on $\mathbf{x}$:

$$\begin{aligned} c_i(\mathbf{x}, \widehat{y}_i, f_i(\mathbf{x}, y_i)) &= \max(1 - \widehat{y}_i(\langle \mathbf{w}_i, \mathbf{x} \rangle + b_i), 0) \\ &= \max(1 - sign(\langle \mathbf{w}_i, \mathbf{x} \rangle + b_i)(\langle \mathbf{w}_i, \mathbf{x} \rangle + b_i), 0) \\ &= \max(1 - |\langle \mathbf{w}_i, \mathbf{x} \rangle + b_i|, 0) \qquad (1 \le i \le q) \end{aligned}$$

For better classification performance on $\mathbf{x}$, we need to minimize the total losses on it, i.e. $\sum_{i=1}^{q} c_i$. Similarly, we also need to minimize the total losses on the whole unlabeled instances in D_u:

$$\min_{\mathbf{W}} \sum_{j=l+1}^{l+u} \sum_{v=1}^{q} \max(1 - |\langle \mathbf{w}_v, \mathbf{x}_j \rangle + b_v|, 0) \tag{3}$$

Eq.(2) can be viewed as regularization framework where the second term corresponds to the loss while the first term corresponds to the regularization term. In that case, we can incorporate Eq.(3) into Eq.(2) as another regularization term which measures the loss caused by unlabeled data. Meanwhile, the class balance constraint is considered to avoid the imbalance prediction on unlabeled instances. Thus we have the optimization problem formulated as follows:

$$\min_{\mathbf{W}, \Xi} \sum_{k=1}^{q} \|\mathbf{w}_k\|^2 + C_1 \sum_{i=1}^{l} \frac{1}{|Y_i||\overline{Y_i}|} \sum_{(y_k, y_l) \in Y_i \times \overline{Y_i}} \xi_{ikl} + C_2 \sum_{j=l+1}^{l+u} \sum_{v=1}^{q} \max(1 - |\langle \mathbf{w}_v, \mathbf{x}_j \rangle + b_v|, 0) \tag{4}$$

$$\begin{aligned}\text{subject to: } & \langle \mathbf{w}_k - \mathbf{w}_l, \mathbf{x}_i \rangle + b_k - b_l \geq 1 - \xi_{ikl} \\ & \xi_{ikl} \geq 0 \qquad (1 \leq i \leq l, (y_k, y_l) \in Y_i \times \overline{Y_i}) \\ & \frac{1}{u} \sum_{j=l+1}^{l+u} \langle \mathbf{w}_v, \mathbf{x}_j \rangle + b_k = \frac{1}{l} \sum_{i=1}^{l} y_{iv} \qquad (1 \leq v \leq q)\end{aligned}$$

where C_1 and C_2 are nonnegative constants that balance the loss on labeled and unlabeled data respectively.

The objective function in Eq.(4) is *non-convex* because the last term consists of the sum of q *non-convex* functions c_i on every unlabeled instance. A learning algorithm can get trapped in the sub-optimal local minimal when trying to find the global minimal solution. In this paper, the ConCave Convex Procedure(*CCCP*) method [Collobert et al. (2006)][Chapelle et al. (2008)] is applied to solve the non-convex optimization problem. In order to apply *CCCP* method on Eq.(4), it is essential to decompose the non-convex function into a convex component and concave component. Here, we re-write the non-convex function as follows:

$$\max(1 - |t|, 0) = \max(1 - |t|, 0) + |t| - |t|$$

in which $t = \langle \mathbf{w}_v, \mathbf{x}_j \rangle + b_v$. If an unlabeled instance $\mathbf{x}_j$ is currently classified positive on label y_v, then at the following iteration, the effective loss on this unlabeled instance will be:

$$\widetilde{L}(t) = \begin{cases} 0 & \text{if } t \geq 1 \\ 1 - t & \text{if } |t| < 1 \\ -2t & \text{if } t \leq -1 \end{cases} \tag{5}$$

A corresponding $\widetilde{L}$ can be defined for the case of an unlabeled instance being classified negative on y_v:

$$\widetilde{L}(t) = \begin{cases} 2t & \text{if } t \geq 1 \\ 1 + t & \text{if } |t| < 1 \\ 0 & \text{if } t \leq -1 \end{cases} \tag{6}$$

Then we can convert Eq.(4) as:

$$\min_{\mathbf{W}, \Xi} \sum_{k=1}^{q} \|\mathbf{w}_k\|^2 + C_1 \sum_{i=1}^{l} \frac{1}{|Y_i||\overline{Y_i}|} \sum_{(y_k, y_l) \in Y_i \times \overline{Y_i}} \xi_{ikl} + C_2 \sum_{j=l+1}^{l+u} \sum_{v=1}^{q} \widetilde{L}(\langle \mathbf{w}_v, \mathbf{x}_j \rangle + b_v) \tag{7}$$

$$\begin{aligned}\text{subject to: } & \langle \mathbf{w}_k - \mathbf{w}_l, \mathbf{x}_i \rangle + b_k - b_l \geq 1 - \xi_{ikl} \\ & \xi_{ikl} \geq 0 \qquad (1 \leq i \leq l, (y_k, y_l) \in Y_i \times \overline{Y_i}) \\ & \frac{1}{u} \sum_{j=l+1}^{l+u} \langle \mathbf{w}_v, \mathbf{x}_j \rangle + b_v = \frac{1}{l} \sum_{i=1}^{l} y_{iv} \qquad (1 \leq v \leq q)\end{aligned}$$

The optimization problem of Eq.(7) is a quadratic programming(*QP*) problem which can be solved efficiently. In summary, Table 1 presents the complete description of *iMLCU*.

Table 2: Statistics of the experimental data sets.

Data set	$\|S\|$	*dim(S)*	*L(S)*	*Lcard(S)*	*LDen(S)*	*DL(S)*	*PDL(S)*	Domain
emotions	593	72	6	1.869	0.311	27	0.046	music
enron	1702	1001	16	2.854	0.178	356	0.209	text
image	2000	294	5	1.236	0.247	20	0.010	images
scene	2407	294	6	1.074	0.179	15	0.006	images
yeast	2417	103	14	4.237	0.303	198	0.082	biology
slashdot	3782	1079	22	1.177	0.054	148	0.039	text
corel5k	5000	499	38	2.090	0.055	894	0.179	text
rcv1-subset1	6000	472	30	2.171	0.072	379	0.063	text
rcv1-subset2	6000	472	30	1.970	0.066	362	0.060	text
rcv1-subset3	6000	472	30	1.953	0.065	347	0.058	text
EURlex-dc	19348	100	41	0.703	0.017	182	0.009	text
EURlex-sm	19348	100	20	1.337	0.067	352	0.018	text

4. Experiments

4.1. Data Set and Evaluation Metrics

To thoroughly evaluate the performance of our approach, a total of twelve real-word multi-label data sets are employed in this paper. For each data set, several statistics are used to depict its characteristics. Specifically, for data set $S = \{(\mathbf{x}_i, Y_i)|1 \leq i \leq p\}$, we denote *number of examples*, *number of features* and *number of possible class labels* as $|S|$,$dim(S)$ and $L(S)$ respectively. In addition, several other specific properties owned by multi-label data [Read et al. (2011)] are denoted as:

• $Lcard(S) = \frac{1}{p}\sum_{i=1}^{p}|Y_i|$: *label cardinality* which measures the average number of labels per example.

• $LDen(S) = \frac{Lcard(S)}{L(S)}$: *label density* which normalizes *LCard(s)* by the number of possible labels.

• $DL(S) = |\{Y|(\mathbf{x}, Y) \in S\}|$: *distinct label sets* which counts the number of distinct label sets in S.

• $PDL(S) = \frac{DL(S)}{|S|}$: *proportion of distinct label sets* which normalizes *DL(S)* by the number of examples.

Table 2 summarizes the detailed statistics of the multi-label data sets used in our experiment in ascending order of $|S|$[1]. For text data sets including enron, corel5k, rcv1 and EURlex, some pre-processing steps are performed including: 1) conducting dimensionality reduction; and 2) filtering *rare* classes. Take text data set rcv1 for example, we keep top 1% frequent words and filter *rare* categories by keeping top 30% frequent categories. Thus we obtain 472 words and 30 topics for every subset of dataset rcv1[2]. As shown in Table 2,

1. In dataset EURlex-dc, there exists many instance without any positive laebl. So *Lcard(S)* of EURlex-dc is less than 1.
2. The reason of reducing dimensionality is to reduce the extremely high computation cost and the reason of filtering categories is to ensure that every label has at least one positive labeled training instance and every labeled training instance has at least one positive label

the twelve data sets cover a broad range of cases whose characteristics are diversified with respect to different multi-label properties.

Performance evaluation in multi-label learning is much complicated than traditional single-label learning, as each example can be associated with multiple labels simultaneously. First, four popular *example-based* multi-label evaluation metrics are employed, i.e. *Ranking Loss*, *One-Error*, *Coverage* and *Average Precision* (Zhang and Zhou (in press)). Briefly, *example-based* metrics evaluate the quality of the predicted label sets for each test example and return the averaged value across all the test examples. Besides, one *label-based* metric is also employed in this paper, i.e. AUC_{macro}. AUC_{macro} evaluates the quality of the predictions for each class label using the AUC criteria and returns the averaged value across all the class labels. For AUC_{macro} and *Average Precision*, the larger the values the better the performance; While for the other three metrics, the smaller the values the better the performance. All these metrics server as good indicators for comprehensive comparisons as they evaluate the performance of algorithms from various perspective.

4.2. Experimental Setup

In this paper, *iMLCU* is compared to four well-established multi-label learning algorithms. Two of them are supervised multi-label algorithms, including *ML-kNN* (Zhang and Zhou (2007)) and *ECC* (Read et al. (2011)). Two of them are semi-supervised multi-label algorithms, including *SMSE* (Chen et al. (2008)) and *TRAM* (Kong et al. (2013)). *ML-kNN* is a *first-order* approach which is derived from the popular k-nearest neighbor technique. Maximum a posterior(MAP) principle is utilized to make prediction by using the statistical information gained from the label sets of a test instance's neighbors. *ECC* is a *high-order* approach. It transforms the multi-label learning problem into a chain of binary classification problems, where subsequent binary classifiers in the chain is built upon the predictions of preceding ones. *SMSE* suggests a regularization framework to combine two graphs as regularizer terms and finally, the algorithm can get the real-value confidences for labels of unlabeled instances by solving a Sylvester equation. *TRAM* introduces the *label concept composition* and the key assumption is similar instances should have similar *label concept composition*. It formulates the transductive multi-label classification as an optimization problem of estimating label concept compositions and derives a closed-form solution to this optimization problem. Both *TRAM* and *SMSE* are transductive algorithms. To the best of our knowledge, none inductive semi-supervised multi-label algorithm has been proposed[3].

For each data set in Table 2, we randomly draw 1% to 5% of the data set as labeled examples and randomly draw 50% of the remaining data set as unlabeled examples. Ten runs of experiments are conducted under every labeled ratio (1% to 5% with stepsize of 1%), and meanwhile the mean value and standard deviation of each evaluation metric are recorded under every label ratio. We denote the set of labeled examples as $\mathbf{L}$ and the set of unlabeled examples as $\mathbf{U}$. The transductive semi-supervised multi-label learning algorithm *TRAM* and *SMSE* train the system on both labeled and unlabeled data and predict the label sets of all the unlabeled data used during training. *iMLCU* is an inductive semi-supervised multi-label learning algorithm which can predict the label sets of the unlabeled

3. In (Sellamanickam et al. (2012)), semi-supervised learning for examples with multiple labels have been studied under the *partial label* setting, i.e. only one of the labels associated to the example is valid.

data not used during training. For fair comparison, we should evaluate the performance of *TRAM*, *SMSE* and *iMLCU* on the same set of test examples. Thus, we extract 20% of the unlabeled examples from $\mathbf{U}$ as test examples, denoted as $\mathbf{T}$. It is obvious that $\mathbf{U} = \mathbf{U'} \cup \mathbf{T}$. In this case, the experiment is implemented as follows:

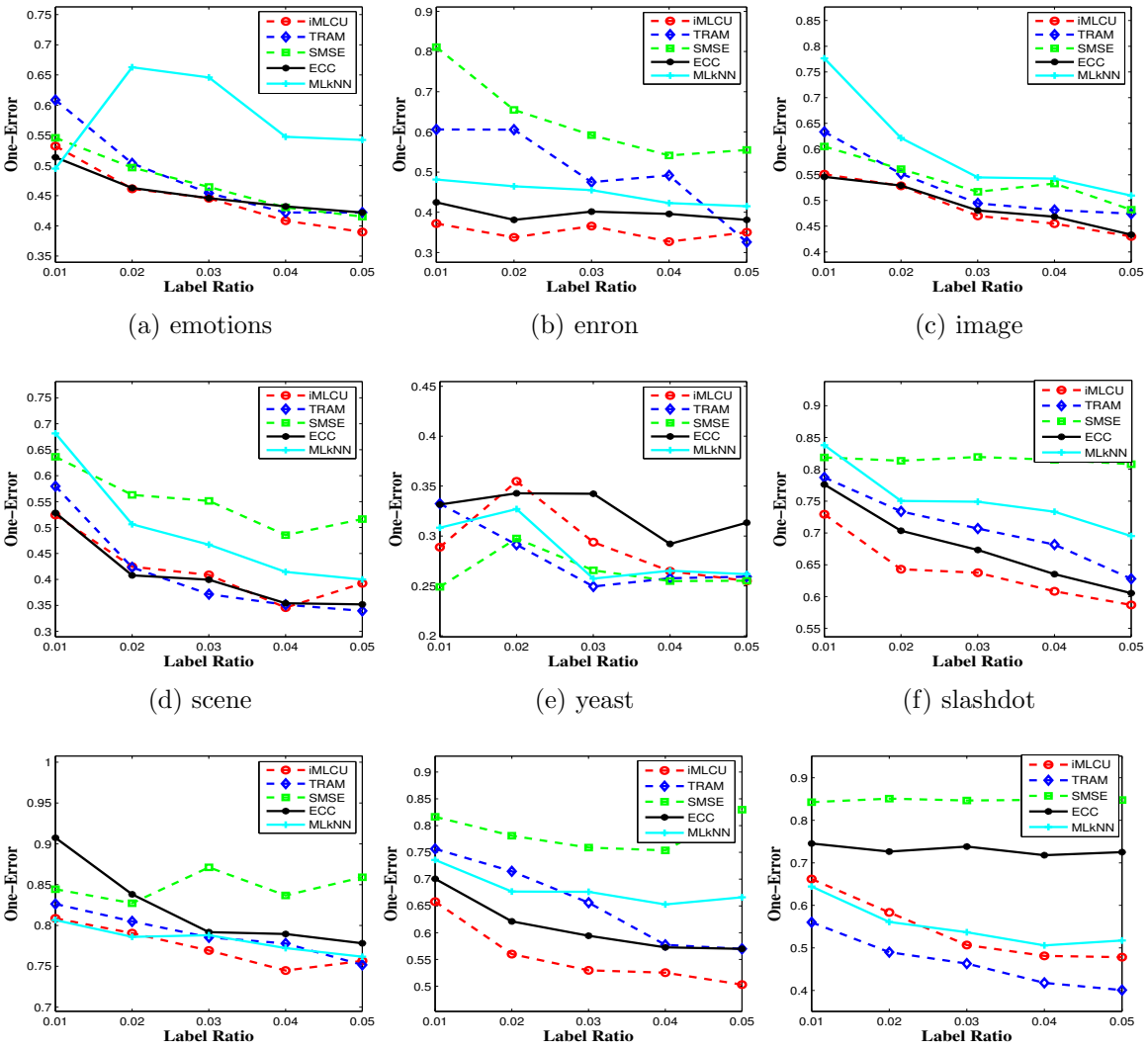
• We learn the system of *TRAM* and *SMSE* on training set $\mathbf{L} \cup \mathbf{U}$ and evaluate the performance on test set $\mathbf{T}$. For *iMLCU*, we learn the system on training set $\mathbf{L} \cup \mathbf{U'}$ and evaluate the performance on test set $\mathbf{T}$. Obviously, the number of training data employed by iMLCU ($\mathbf{L} \cup \mathbf{U'}$) is less than that employed by *TRAM* and *SMSE* ($\mathbf{L} \cup \mathbf{U}$). For the fully supervised algorithms *ECC* and *ML-kNN*, they are trained on the labeled data set $\mathbf{L}$ and tested on the test set $\mathbf{T}$.

ECC is implemented upon MULAN library (Tsoumakas et al. (2011)) while the other four algorithms are implemented in MATLAB. Parameters suggested in respective literatures are adopted for the comparing algorithms unless other specified. For algorithm *SMSE*, we use fully connected graph instead of kNN graph, which is adopted in the original literature. It is believable that this can improve the performance of the algorithm *SMSE*. For *iMLCU*, the parameters needed to be specified are C_1 and C_2 as shown in Eq.(7). In preliminary experiments, cross validation is conducted on some data sets by varying C_1 and C_2 from 0.001 to 100 with scale of 10. Results show that *iMLCU* yields stable performance with $C_1 = 20$ and $C_2 = 0.01$, which are used for iMLCU in this paper. So C_1 and C_2 are set to be 20 and 0.01 respectively for all the data sets in this paper. Furthermore, as shown in Table 1, the maximum number of iterations (*maxIter*) is set to 20, and the optimization procedure is deemed to be converged if the value of the objective function in Eq.(7) does not decrease significantly after each iteration (vary less than 1 percent).

4.3. Experimental Results

Due to space limitation, instead of all the five evaluation criteria we only illustrate the experimental results of three evaluation criteria on nine data sets (excluding rcv1-subset2, rcv1-subset3 and Eurlex-sm for brevity), i.e. *One-Error*, *Average Precision* and AUC_{macro}, in Figure 1 to Figure 3 respectively. On *example-based* evaluation criteria *One-Error* and *Average Precision*, *iMLCU* achieves better or at least comparable classification performance against other four comparing algorithms over almost every data set. On *label-based* evaluation criteria AUC_{macro}, *iMLCU* and *TRAM* obviously outperform other three algorithms and achieve comparable performance over most data sets. Under each labeled ratio, we have 60 configurations for comparison (12 data sets x 5 criteria) against each comparing algorithm. Generally, under labeled ratios 1% to 5%, *iMLCU* ranks *1st* in 50%, 50%, 48.3%, 31.7%, and 28.3% cases, ranks *2nd* in 23.3%, 28.3%, 26.7%, 40%, and 45% cases, and never ranks *5th* except for the 3.3% cases when labeled ratio is only 2%. It is noticeable that the case of *iMLCU* ranking *1st* increases as the labeled ratio decreases, which indicates that our approach can handle the situation of few labeled data well.

To perform statistical comparative analysis, under each labeled ratio, *paired t-test* is further conducted which compares *iMLCU* with other algorithms on each data set with respect to every criteria. Table 3 summarizes the detailed results of statistical comparison. From Table 3, we can conclude that our approach outperforms the two supervised algorithms



(a) emotions (b) enron (c) image

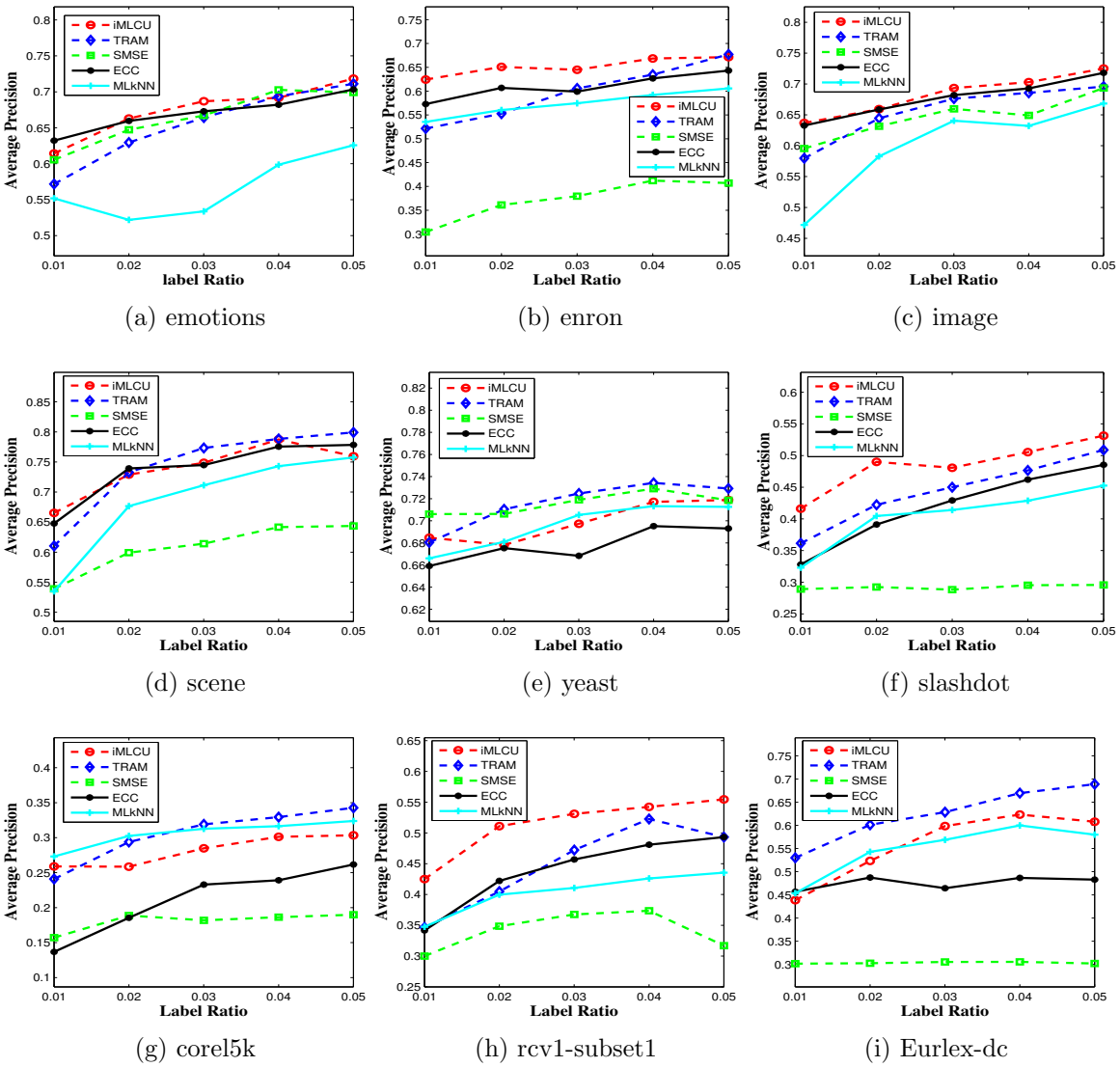
(d) scene (e) yeast (f) slashdot

(g) corel5k (h) rcv1-subset1 (i) Eurlex-dc

Figure 1: Experimental results on the nine data sets in terms of *One-Error*, where x-axis is label ratio and y-axis is *One-Error* value. The lower the curve, the better the performance.

ECC and *ML-kNN* on every evaluation criteria, which indicates that *iMLCU* does have the ability of combining unlabeled data with labeled ones to help improve generalization performance.

With respect to semi-supervised multi-label learning algorithms, it is notable that *iMLCU* outperforms the *SMSE* on every evaluation criteria. In terms of AUC_{macro}, *TRAM* achieves better performance than *iMLCU* and with the increase of labeled data, the performance of *TRAM* on AUC_{macro} is getting better. Note that TRAM has an extra *embedding* dimensionality reduction strategy which is shown to be essential for achieving good performance (Kong et al. (2013)), while no such strategy is employed by *iMLCU*. Furthermore,



(a) emotions (b) enron (c) image

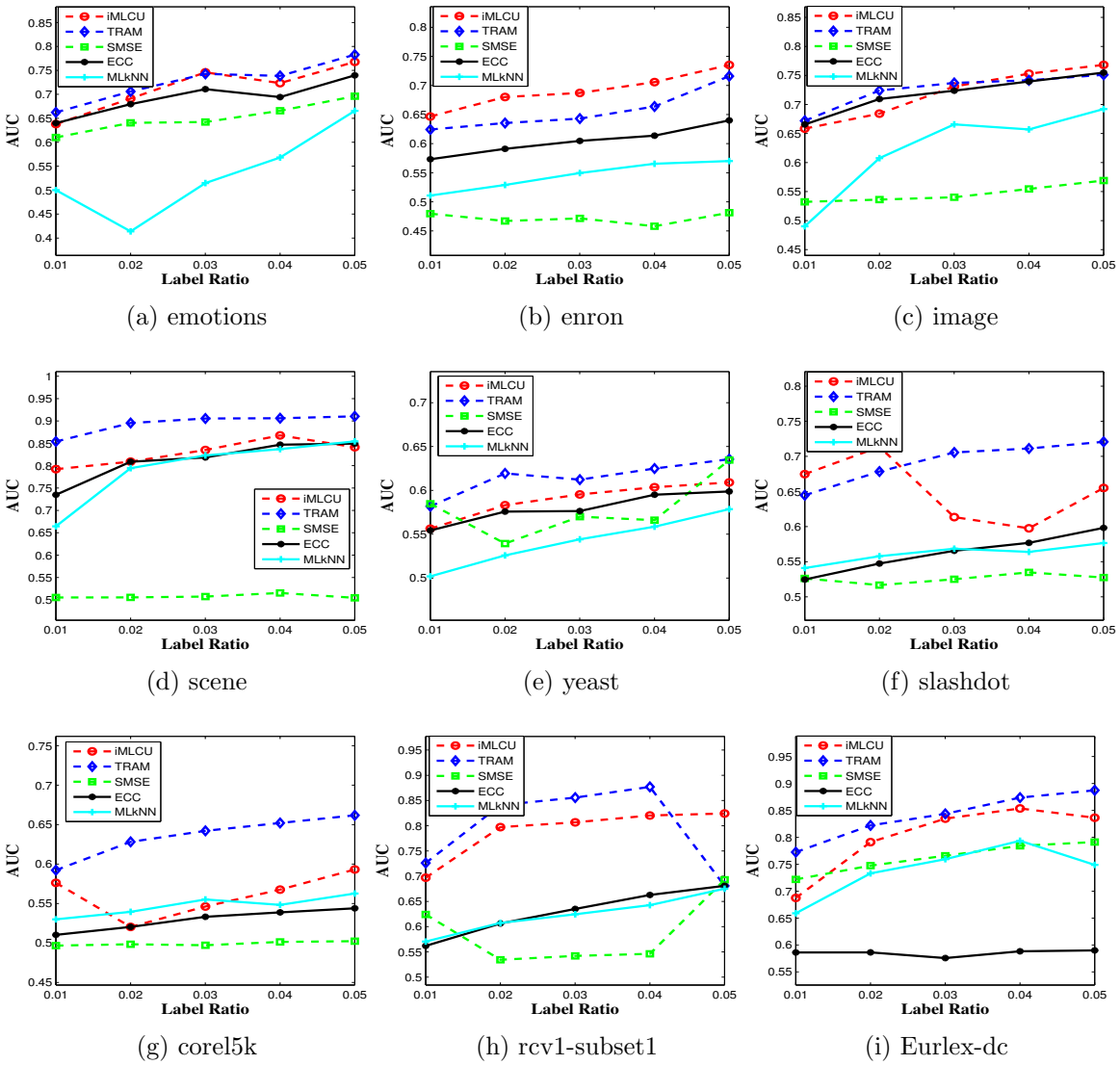
(d) scene (e) yeast (f) slashdot

(g) corel5k (h) rcv1-subset1 (i) Eurlex-dc

Figure 2: Experimental results on nine data sets in terms of *Average Precision*, where x-axis is label ratio and y-axis is *Average Precision* value. The higher the curve, the better the performance.

as stated in Section 4.2, more unlabeled data (i.e. **U**) have been utilized by TRAM in the training phase than those (i.e. **U'**) utilized by *iMLCU*. On the other evaluation criteria, *iMLCU* performs favorably against TRAM.

Note that our approach can also work under the transductive setting, i.e. to predict the label sets of unlabeled data used during training like *TRAM* and *SMSE*. Under transductive setting, *iMLCU*,*TRAM* and *SMSE* train their systems on training set $\mathbf{L} \cup \mathbf{U'}$ and evaluate the performance on **U'**, where **L** and **U'** are defined in Section 4.2. Complementary to the inductive experiments, we also compare the performance of the semi-supervised algorithms



(a) emotions (b) enron (c) image

(d) scene (e) yeast (f) slashdot

(g) corel5k (h) rcv1-subset1 (i) Eurlex-dc

Figure 3: Experimental results on the nine data sets in terms of AUC_{macro}, where x-axis is label ratio and y-axis is AUC_{macro} value. The higher the curve, the better the performance.

under transductive setting. Due to space limitation, detailed results on four representative data sets are shown in Table 4. The best performance among the three comparing algorithms is highlighted in boldface. For each evaluation criterion, "$\downarrow$" indicates the "the smaller the better" while "$\uparrow$" indicates "the larger the better". As shown in Table 4, it is impressive that in most cases *iMLCU* achieves competitive results against *TRAM* and *SMSE*.

To show the scalability of the proposed approach, we also study the training time required by *iMLCU* as the number of unlabeled data and the number of class labels increases respectively. Due to space limitation, Figure 4 only reports the results on data set *corel5k* with different labeled ratios (LR=1% to 5%) for illustrative purpose. Specifically, the x-axis

Table 3: Paired t-test result(win/tie/lose) over the twelve datasets when comparing *iMLCU* with other four algorithms.

Label Ratio	Evaluation Metric	*iMLCU* versus			
		ECC	*ML-kNN*	*TRAM*	*SMSE*
1%	Ranking Loss	8/2/2	8/4/0	8/1/3	10/1/1
	One-Error	7/5/0	7/5/0	6/4/2	9/3/0
	Coverage	10/2/0	8/3/1	7/2/3	10/1/1
	Average Precision	7/4/1	9/2/1	7/3/2	10/1/1
	AUC_{macro}	9/3/0	12/0/0	2/3/7	8/2/2
2%	Ranking Loss	8/4/0	9/1/2	6/3/3	9/2/1
	One-Error	8/4/0	9/3/0	6/3/3	8/3/1
	Coverage	9/3/0	8/2/2	6/2/4	9/2/1
	Average Precision	7/5/0	9/2/1	6/2/4	9/2/1
	AUC_{macro}	6/4/2	10/1/1	3/0/9	12/0/0
3%	Ranking Loss	12/0/0	8/3/1	3/3/6	11/0/1
	One-Error	8/4/0	10/1/1	5/3/4	10/1/1
	Coverage	12/0/0	9/1/2	4/2/6	11/0/1
	Average Precision	9/3/0	10/2/0	5/2/5	10/1/1
	AUC_{macro}	10/2/0	10/2/0	1/4/7	12/0/0
4%	Ranking Loss	11/1/0	7/1/4	1/4/7	10/0/2
	One-Error	8/4/0	10/2/0	5/5/2	10/2/0
	Coverage	10/2/0	7/1/4	1/6/5	10/0/2
	Average Precision	10/2/0	9/1/2	3/4/5	10/1/1
	AUC_{macro}	9/3/0	12/0/0	1/2/9	12/0/0
5%	Ranking Loss	9/3/0	6/3/3	2/2/8	10/1/1
	One-Error	8/3/1	8/3/1	3/5/4	10/2/0
	Coverage	9/3/0	6/3/3	2/3/7	10/1/1
	Average Precision	9/2/1	9/2/1	3/3/6	10/2/0
	AUC_{macro}	11/1/0	11/1/0	3/1/8	11/0/1

in Figure 4(a) corresponds to the number of unlabeled data used in training, while that in Figure 4(b) corresponds to the number of class labels being considered in training. As shown in Figure 4, the training time required by *iMLCU* scales well (being nearly linear) as the complexity of the learning problem increases.

5. Conclusion

In this paper, the problem of inductive semi-supervised learning for multi-label data has been studied. To the best of our knowledge, the proposed *iMLCU* approach is the first attempt toward inductive-style semi-supervised multi-label learning. By considering pairwise label correlations over labeled data and imposing maximum-margin regularization over unlabeled data, *iMLCU* induces a collection of linear models via the iterative *CCCP* procedure. Experimental results on a total of twelve benchmark data sets clearly validate the good performance of *iMLCU* on learning from both labeled and unlabeled multi-label data.

In the future, it is interesting to see whether the optimization problem of *iMLCU* could be formulated in other ways such as considering different forms of label correla-

Table 4: Transductive experimental results(mean) on every label ratio.

Label Ratio	Data Set	Algorithms	Ranking Loss↓	One-Error↓	Coverage↓	Average Precision↑	AUC_{macro} ↑
1%	enron	*iMLCU*	**0.2171**	**0.3813**	**7.031**	**0.6121**	**0.6532**
		TRAM	0.2654	0.6442	7.581	0.5272	0.6091
		SMSE	0.5567	0.7078	10.87	0.3329	0.4755
	image	*iMLCU*	**0.3265**	**0.5383**	**1.561**	**0.6429**	0.6698
		TRAM	0.3814	0.6155	1.771	0.5902	**0.6793**
		SMSE	0.3685	0.5750	1.698	0.6145	0.5341
	rcv1-subset1	*iMLCU*	**0.2403**	**0.6598**	**11.05**	**0.4228**	0.6957
		TRAM	0.2797	0.7401	13.05	0.3576	**0.7203**
		SMSE	0.3294	0.8098	14.33	0.3057	0.6256
	rcv1-subset2	*iMLCU*	**0.2330**	**0.6100**	**10.07**	**0.4550**	0.6926
		TRAM	0.2639	0.6820	11.42	0.4037	**0.7184**
		SMSE	0.3145	0.7282	12.78	0.3549	0.6282
2%	enron	*iMLCU*	**0.1976**	**0.3569**	**6.739**	**0.6418**	**0.6793**
		TRAM	0.2332	0.5691	7.185	0.5638	0.6362
		SMSE	0.5293	0.7013	10.58	0.3540	0.4633
	image	*iMLCU*	**0.2881**	**0.4974**	**1.412**	**0.6743**	0.6955
		TRAM	0.3016	0.5261	1.466	0.6579	**0.7308**
		SMSE	0.3330	0.5533	1.572	0.6349	0.5349
	rcv1-subset1	*iMLCU*	**0.1749**	**0.5567**	**8.563**	**0.5115**	0.7943
		TRAM	0.2250	0.7193	10.84	0.4071	**0.8388**
		SMSE	0.2825	0.7837	11.98	0.3483	0.5349
	rcv1-subset2	*iMLCU*	**0.1784**	**0.5446**	**8.009**	**0.5218**	0.7810
		TRAM	0.1991	0.5969	9.107	0.4870	**0.8421**
		SMSE	0.2826	0.7510	11.00	0.3695	0.5498
3%	enron	*iMLCU*	**0.1954**	**0.3718**	**6.665**	**0.6404**	**0.6881**
		TRAM	0.2063	0.4559	6.740	0.6083	0.6542
		SMSE	0.5283	0.7255	10.44	0.3556	0.4592
	image	*iMLCU*	**0.2787**	**0.4831**	**1.383**	**0.6827**	0.7167
		TRAM	0.2909	0.5079	1.425	0.6682	**0.7308**
		SMSE	0.3182	0.5241	1.516	0.6536	0.5352
	rcv1-subset1	*iMLCU*	**0.1641**	**0.5305**	**8.180**	**0.5323**	0.8087
		TRAM	0.1820	0.6450	9.267	0.4771	**0.8594**
		SMSE	0.2642	0.7525	11.38	0.3678	0.5417
	rcv1-subset2	*iMLCU*	0.1621	**0.5271**	**7.472**	**0.5454**	0.7975
		TRAM	**0.1540**	0.5598	7.674	0.5422	**0.8617**
		SMSE	0.2537	0.6883	10.20	0.4100	0.5749
4%	enron	*iMLCU*	**0.1915**	**0.3555**	**6.509**	**0.6508**	**0.6999**
		TRAM	0.1921	0.4536	6.572	0.6358	0.6645
		SMSE	0.5324	0.6552	10.60	0.3594	0.4588
	image	*iMLCU*	**0.2572**	**0.4450**	**1.299**	**0.7070**	0.7442
		TRAM	0.2607	0.4693	1.318	0.6945	**0.7514**
		SMSE	0.2971	0.5132	1.438	0.6654	0.5530
	rcv1-subset1	*iMLCU*	0.1563	**0.5158**	7.856	**0.5486**	0.8170
		TRAM	**0.1430**	0.5620	**7.656**	0.5395	**0.8789**
		SMSE	0.2441	0.7455	10.67	0.3830	0.5467
	rcv1-subset2	*iMLCU*	0.1582	**0.5153**	7.299	0.5575	0.8057
		TRAM	**0.1314**	0.5320	**6.832**	**0.5788**	**0.8762**
		SMSE	0.2437	0.6923	9.746	0.4168	0.5781
5%	enron	*iMLCU*	0.1751	**0.3336**	**6.118**	**0.6769**	**0.7281**
		TRAM	**0.1734**	0.3360	6.159	0.6732	0.7093
		SMSE	0.4877	0.6012	10.40	0.4100	0.4871
	image	*iMLCU*	**0.2338**	**0.4108**	**1.208**	**0.7303**	**0.7637**
		TRAM	0.2487	0.4572	1.268	0.7051	0.7607
		SMSE	0.2740	0.4824	1.338	0.6887	0.5693
	rcv1-subset1	*iMLCU*	**0.1545**	**0.5030**	**7.789**	**0.5542**	**0.8215**
		TRAM	0.1734	0.6254	8.888	0.4811	0.8005
		SMSE	0.3034	0.8262	13.22	0.3186	0.6947
	rcv1-subset2	*iMLCU*	0.1517	**0.5062**	7.115	0.5674	0.8152
		TRAM	**0.1159**	0.5131	**6.137**	**0.5997**	**0.8825**
		SMSE	0.2333	0.6992	9.570	0.4179	0.5650

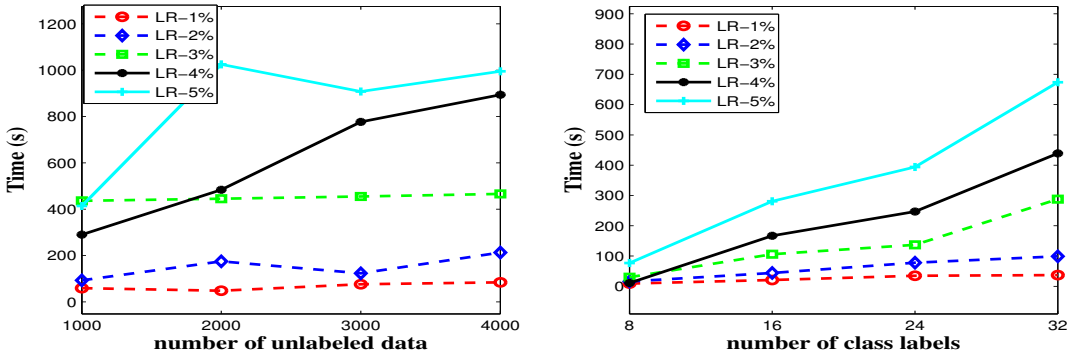


Figure 4: Training time of *iMLCU* on data set *corel5k* with: (a) increasing number of unlabeled data; (b) increasing number of class labels.

tions. Furthermore, designing other strategies for accomplishing inductive semi-supervised multi-labeling is also worth further study.

References

O. Chapelle, V. Sindhwaniand, and S.-S. Keerthi. Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9:203–233, 2008.

G. Chen, Y.-Q. Song, F. Wang, and C.-S. Zhang. Semi-supervised multi-label learning by solving a sylvester equation. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 410–419, Atlanta, GA, 2008.

R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large scale transductive svms. *Journal of Machine Learning Research*, 7:1687–1712, 2006.

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

J. Fürnkranz, E. Hüllermeier, E.-L. Mencía, and K. Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153, 2008.

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

Y.-H. Guo and D. Schuurmans. Semi-supervised multi-label classification: a simultaneous large-margin, subspace learning approach. In P.-A. Flach, T.-D. Bie, and N. Cristianini, editors, *Lecture Notes in Computer Science 7524*, pages 355–370. Berlin: Springer, Bristol, UK, 2012.

T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of 16th International Conference on Machine Learning*, pages 200–209, San Francisco, CA, 1999.

X.-N. Kong, M. Ng, and Z.-H. Zhou. Transductive multi-label learning via label set propagation. *IEEE Transactions on Knowledge and Data Mining*, 25(3):704–719, 2013.

J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011.

S. Sellamanickam, C. Tiwari, and S.-K. Selvaraj. Regularized structured output learning with partial labels. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 1059–1070, Anaheim, CA, 2012.

G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining multi-label data. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 667–686. Berlin: Springer, 2010.

G. Tsoumakas, E.-S. Xioufis, J. Vilcek, and I.-P. Vlahavas. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12(7):2411–2414, 2011.

J.-D. Wang, Y.-H. Zhao, X.-Q. Wu, and X.-S. Hua. A transductive multi-label learning approach for video concept detection. *Pattern Recognition*, 44(10):2274–2286, 2011.

Z.-J. Zha, T. Mei, J.-D. Wang, Z.-F. Wang, and X.-S. Hua. Graph-based semi-supervised learning with multiple labels. *Journal of Visual Communication and Image Representation*, 20(2):97–103, 2009.

M.-L. Zhang and Z.-H. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.

M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, in press.

D.-Y. Zhou, O. Bousquet, TN. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pages 321–328. 2004.

S.-H. Zhu, X. Ji, W. Xu, and Y.-H. Gong. Multi-labelled classification using maximum entropy method. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 274–281, Salvador, Brazil, 2005.

X.-J. Zhu and A.-B. Goldberg. Introduction to semi-supervised learning. In R. Brachman and T. Dietterich, editors, *Synthesis Lectures on Artificial Intelligence and Machine Learning*, pages 1–130. Maogen and Claypool, 2009.

X.-J. Zhu, Z.-B. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of 20th International Conference on Machine Learning*, pages 912–919, Wanshington D.C, 2003.