Inductive Semi-supervised Multi-Label Learning with Co-Training

Wang Zhan[1,2], Min-Ling Zhang[1,2,3,*]
[1]School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
[2]Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China
[3]Collaborative Innovation Center of Wireless Communications Technology, China
{zhanw,zhangml}@seu.edu.cn

ABSTRACT

In multi-label learning, each training example is associated with multiple class labels and the task is to learn a mapping from the feature space to the *power set* of label space. It is generally demanding and time-consuming to obtain labels for training examples, especially for multi-label learning task where a number of class labels need to be annotated for the instance. To circumvent this difficulty, semi-supervised multi-label learning aims to exploit the readily-available unlabeled data to help build multi-label predictive model. Nonetheless, most semi-supervised solutions to multi-label learning work under transductive setting, which only focus on making predictions on existing unlabeled data and cannot generalize to unseen instances. In this paper, a novel approach named Coins is proposed to learning from labeled and unlabeled data by adapting the well-known co-training strategy which naturally works under *inductive* setting. In each co-training round, a dichotomy over the feature space is learned by maximizing the diversity between the two classifiers induced on either dichotomized feature subset. After that, pairwise ranking predictions on unlabeled data are communicated between either classifier for model refinement. Extensive experiments on a number of benchmark data sets show that Coins performs favorably against state-of-the-art multi-label learning approaches.

CCS CONCEPTS

• **Computing methodologies** → *Semi-supervised learning settings*; *Machine learning approaches*;

1 INTRODUCTION

Multi-label learning deals with the problem where each example is represented by a single instance (feature vector) while associated with multiple class labels simultaneously [12, 29]. Correspondingly, the task is to learn a multi-label predictor which maps from the input space of instances to the output space of *label sets*. Due to the huge size of output space (exponential to the number of class labels), significant number of labeled training examples are needed in order to build multi-label predictor with good generalization performance. Nonetheless, the process of obtaining labels for training examples is generally demanding and time-consuming, especially for multi-label data where more than one class label should be annotated. Therefore, a natural remedy is to consider semi-supervised multi-label learning which makes use of the readily-available unlabeled data to help build the predictive model [7, 14, 15, 17, 21, 22, 26, 28].

Nonetheless, most attempts towards semi-supervised multi-label learning work under the transductive setting, which only focus on classifying given unlabeled data and thus cannot generalize to unseen instances. Generally, graph-based semi-supervised techniques are utilized to construct an affinity matrix over both labeled and unlabeled data, where classifications on unlabeled data can be obtained via label propagation [17, 21] or manifold regularization [7, 15, 22, 28]. In many cases, however, it is obviously more desirable to endow learning system with the *inductive* ability of making predictions on unseen instances other than existing labeled and unlabeled data.

Different to graph-based semi-supervised learning techniques, the well-known *co-training* strategy trains two classifiers over two feature sets which are iteratively updated by communicating either classifier's predictions on unlabeled data to augment the labeled training set of the other [4, 30, 31]. Co-training offers a natural way to enable inductive semi-supervised learning where the outputs of both classifiers on unseen instances can be combined to yield the final predictions. To amend co-training for multi-label learning task, the key adaptation lies in how to generate two classifiers based on the original feature space and how to achieve effective supervision information communication between the classifiers.

In light of the above analysis, a novel approach named Coins, i.e. *CO-training for INductive Semi-supervised multi-label learning*, is proposed in this paper. In each co-training round, the original feature space is automatically dichotomized by maximizing the diversity between the two classifiers induced on either dichotomized feature subset. After that, pairwise ranking predictions on unlabeled data are communicated between the two classifiers for model refinement. Experiments on a number of benchmark data sets clearly validate the effectiveness of conducting semi-supervised multi-label learning with the co-training strategy.

The rest of this paper is organized as follows. Section 2 briefly reviews related work on semi-supervised multi-label learning. Section 3 presents the technical details of Coins. Section 4 reports

*Corresponding author

KDD'17, August 13-17, 2017, Halifax, NS, Canada

ACM ISBN 978-1-4503-4887-4/17/08…$15.00
https://doi.org/10.1145/3097983.3098141

experimental results of comparative studies. Finally, Section 5 concludes.

2 RELATED WORK

Let $\mathcal{X} = \mathbb{R}^d$ denote the d-dimensional feature space and $\mathcal{Y} = \{\lambda_1, \lambda_2, \ldots, \lambda_q\}$ denote the label space consisting of q class labels. The task of multi-label learning is to induce a multi-label predictive model $h : \mathcal{X} \mapsto 2^{\mathcal{Y}}$ from the training examples, which aims to assign *a set of* relevant labels $h(\boldsymbol{x}) \subseteq \mathcal{Y}$ for an instance $\boldsymbol{x} \in \mathcal{X}$. In the following, related work on semi-supervised multi-label learning will be briefly reviewed while comprehensive introductions on multi-label learning in the general setting can be found in [12, 29].

In semi-supervised learning setting, in addition to L labeled training data $\mathcal{L} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_L, \boldsymbol{y}_L)\}$, a repository of U unlabeled data $\mathcal{U} = \{\boldsymbol{x}_{L+1}, \ldots, \boldsymbol{x}_{L+U}\}$ are also available to the learning system. Here, $\boldsymbol{x}_i \in \mathcal{X}$ is a d-dimensional feature vector $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}]^\top$ and $\boldsymbol{y}_i \in \{+1, -1\}^q$ is a q-bits label vector $\boldsymbol{y}_i = (y_{i1}, y_{i2}, \ldots, y_{iq})$ with $y_{ik} = +1$ (-1) indicating whether λ_k is a relevant (irrelevant) label for $\boldsymbol{x}_i$. Generally, the number of labeled data is much less than the number of unlabeled data, i.e. $L \ll U$.

To learn from both $\mathcal{L}$ and $\mathcal{U}$, a common practice is to rely on graph-based semi-supervised learning techniques [31]. Specifically, an $(L+U) \times (L+U)$ affinity matrix $\mathbf{M}$ is built over all the labeled and unlabeled data, where $\mathbf{M}(i, j)$ specifies the similarity degree between instances $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Thereafter, the label vector $\boldsymbol{y}_i$ on unlabeled instance $\boldsymbol{x}_i$ $(L+1 \le i \le L+U)$ can be estimated by conducting iterative label propagation based on $\mathbf{M}$ [17, 21], where in the t-th iteration $\boldsymbol{y}_i(t)$ is updated with labeling information propagated from other instances $\boldsymbol{x}_j$ weighted by $\mathbf{M}(i, j)$, e.g. $\boldsymbol{y}_i(t) \propto \sum_{j=1}^{L+U} \mathbf{M}(i, j) \cdot \boldsymbol{y}_j(t-1)$. Alternatively, $\boldsymbol{y}_i$ can be estimated by enforcing manifold regularization based on $\mathbf{M}$ [7, 15, 22, 28], where similar instances are assumed to have similar labeling vectors, e.g. $||\boldsymbol{y}_i - \boldsymbol{y}_j|| \propto ||\boldsymbol{x}_i - \boldsymbol{x}_j||/\mathbf{M}(i, j)$.

Obviously, graph-based techniques are only capable of making predictions on available unlabeled data $\mathcal{U}$ while cannot generalize to unseen instance. In [14], transductive semi-supervised learning is performed by utilizing unlabeled data in another way, where a low-dimensional subspace mapping for the original feature space is learned from both $\mathcal{L}$ and $\mathcal{U}$ such that the predictive model is induced from the mapped labeled training data. Semi-supervised multi-label learning techniques have also been found useful in a number of applications such as affective computing [18], image processing [19, 25], dimensionality reduction [5, 27], etc.

In [26], an inductive semi-supervised learning approach named iMLCU is proposed by adapting the semi-supervised support vector machines (S3VM) [16, 31]. The resulting optimization problem consists of empirical loss term on labeled data and regularization term on unlabeled data, which is non-convex and optimized by the concave convex procedure (CCCP) [6, 9]. Similar to S3VM, iMLCU specifies the regularization term by treating unreliable predictions on unlabeled data as *pseudo* labels to measure the loss on unlabeled data.

In the next section, a novel semi-supervised multi-label learning approach is presented which works inductively by adapting the well-known co-training strategy.

3 THE PROPOSED APPROACH

To enable co-training for the task of learning from multi-label data, two key adaptations need to be instantiated accordingly: 1) Generation of two classifiers from both $\mathcal{L}$ and $\mathcal{U}$ based on the original feature space $\mathcal{X}$; 2) Supervision information communication for classifier update.

Standard co-training techniques apply to data with two views which satisfy the *sufficient* and *independent* conditions, i.e. each view contains sufficient information to induce a classifier with strong generalization ability and is independent to each other given the class label. Existing studies show that single-view co-training can be also be successful [1, 8, 11, 13], and theoretical studies disclose that the key for co-training to succeed lies the existence of large diversity between the two classifiers [23, 24]. In light of the success of view splitting in designing single-view co-training approaches, COINS employs similar strategies [8, 11] to generate multi-label predictive model.

Let $\mathbf{W} = \{\boldsymbol{w}_k \mid 1 \le k \le q\}$ denote the multi-label (linear) classification models to be learned from $\mathcal{L} \bigcup \mathcal{U}$, where $\boldsymbol{w}_k = [w_{k1}, w_{k2}, \ldots, w_{kd}]^\top \in \mathbb{R}^d$ is the weight vector corresponding to the k-th class label λ_k.[1] In this paper, COINS makes use of *ranking loss* for classifier induction which has been widely-used to develop effective multi-label learning approaches [12, 29]. Specifically, ranking loss considers the ranking relationship between a pair of class labels where the modeling output on relevant label should be larger than that on irrelevant label. Let $\Omega = \{(i, k, l) \mid 1 \le i \le L+U, 1 \le k \ne l \le q\}$ denote the set of indexing triplets for all possible instances and label pairs. Given a chosen index set $\mathcal{I} \subseteq \Omega$, the empirical ranking loss of classification model $\mathbf{W}$ w.r.t. $\mathcal{I}$ corresponds to:

$$RL(\mathbf{W}, \mathcal{I}) = \frac{1}{|\mathcal{I}|} \cdot \sum_{(i,k,l) \in \mathcal{I}} \max(0, 1 - \langle \boldsymbol{w}_k - \boldsymbol{w}_l, \boldsymbol{x}_i \rangle) \tag{1}$$

Here, $\langle \cdot, \cdot \rangle$ represents the inner product between two feature vectors and $|\cdot|$ returns the set cardinality. Furthermore, the loss to be optimized for the classification model usually considers the balance between empirical ranking loss and model complexity:

$$V(\mathbf{W}, \mathcal{I}) = \frac{1}{2} \sum_{k=1}^{q} ||\boldsymbol{w}_k||^2 + C \cdot RL(\mathbf{W}, \mathcal{I}) \tag{2}$$

Here, C represents the balancing parameter.

3.1 Generation of Classifiers

COINS aims to the learn two classification models $\mathbf{W}^1$ and $\mathbf{W}^2$ from labeled and unlabeled data, where the corresponding index sets $\mathcal{I}^1$ and $\mathcal{I}^2$ can be initialized based on the labeled training data $\mathcal{L} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) \mid 1 \le i \le L\}$:

$$\mathcal{I}^1 = \mathcal{I}^2 = \{(i, k, l) \mid 1 \le i \le L, y_{ik} = +1, y_{il} = -1\} \tag{3}$$

To ensure good performance of both classification models, $\mathbf{W}^1$ and $\mathbf{W}^2$ can be jointly trained by minimizing the following objective function:

$$\min_{\{\mathbf{W}^1, \mathbf{W}^2\}} \max\left(V(\mathbf{W}^1, \mathcal{I}^1), V(\mathbf{W}^2, \mathcal{I}^2)\right) \tag{4}$$

[1] WLOG, the bias value can be absorbed into $\boldsymbol{w}_k$ as well by adding an extra feature with constant value 1.

By replacing the non-differentiable max operator with its softmax relaxation, Eq.(4) can be rewritten as:

$$\min_{\{\mathbf{W}^1,\mathbf{W}^2\}} \log\left(e^{V(\mathbf{W}^1,\mathcal{I}^1)}+e^{V(\mathbf{W}^2,\mathcal{I}^2)}\right) \tag{5}$$

Note that Eq.(5) needs to be optimized by considering that the diversity between $\mathbf{W}^1$ and $\mathbf{W}^2$ should be large to enable effective supervision information communication between them. As discussed previously, Coins employs the view splitting strategy to generate multi-label predictive model. A dichotomy over the original feature space $\mathcal{X}$ can be specified equivalently by enforcing the following constraint over $\mathbf{W}^1$ and $\mathbf{W}^2$:

$$\sum_{a=1}^{d} \left(w_{ka}^1\right)^2 \cdot \left(w_{ka}^2\right)^2 = 0 \ (\forall\ 1 \le k \le q) \tag{6}$$

The above constraint ensures that at each dimension either $\boldsymbol{w}_k^1$ or $\boldsymbol{w}_k^2$ will have a zero weight value. Furthermore, the above constraint is enforced w.r.t. each class label λ_k which offers the flexibility of obtaining tailored dichotomy for different class label.

Along with the dichotomy over feature space, Coins adapts the popular ϵ-*expansion* property of co-training to characterize the diversity between two classification models [2, 8]. Roughly speaking, let $\mathfrak{D}$ be the underlying distribution from which labeled examples are drawn and S^1 (S^2) be the event that an instance in $S \subseteq \mathcal{X}$ is classified *confidently* by the first (second) classifier. Let $\Pr(\mathrm{S}^1 \wedge \mathrm{S}^2)$, $\Pr(\mathrm{S}^1 \oplus \mathrm{S}^2)$ and $\Pr(\overline{\mathrm{S}^1} \wedge \overline{\mathrm{S}^2})$ denote the probability that both S^1 and S^2 hold, exactly one of S^1 and S^2 holds, and none of S^1 and S^2 hold respectively. Then, the ϵ-expansion property is said to be satisfied for $\mathfrak{D}$ if the following condition is met w.r.t. any S and classifiers within the hypothesis class:

$$\Pr(\mathrm{S}^1 \oplus \mathrm{S}^2) \ge \epsilon \cdot \min\left[\Pr(\mathrm{S}^1 \wedge \mathrm{S}^2), \Pr(\overline{\mathrm{S}^1} \wedge \overline{\mathrm{S}^2})\right] \tag{7}$$

To make use of the ϵ-expansion property, Coins chooses to indicate the confidence of multi-label classification by the model's predictive difference over a pair of class labels. Specifically, given a pair of class labels (y_k, y_l) $(k \ne l)$, the *confidence indicator* function of the classification model $\mathbf{W}$ on an instance $\boldsymbol{x}$ is defined as:

$$c_{\mathbf{W}}^{k,l}(\boldsymbol{x}) = \frac{1}{1+\exp\left(-\gamma \cdot (p_{\mathbf{W}}^{k,l}(\boldsymbol{x}) - \tau)\right)} \tag{8}$$

Here, $p_{\mathbf{W}}^{k,l}(\boldsymbol{x}) = (1+\exp(-\langle \boldsymbol{w}_k - \boldsymbol{w}_l, \boldsymbol{x}\rangle))^{-1}$ gives the confidence that on instance $\boldsymbol{x}$ the classification model $\mathbf{W}$ yields a higher rank for y_k than y_l. Furthermore, the parameter τ in Eq.(8) controls how significant the confidence should be in order to trigger the indicator function. In this paper, τ is set to be 0.8 following [8]. Furthermore, γ is set to be 50 to keep the sigmoid function Eq.(8) steep and thus serve as a good approximation of the 0-1 indicator function.

Correspondingly, the ϵ-expansion property can be empirically realized based on the unlabeled data in $\mathcal{U}$:

$$\sum_{\boldsymbol{x}\in\mathcal{U}} \left[c_{\mathbf{W}^1}^{k,l}(\boldsymbol{x}) \cdot \bar{c}_{\mathbf{W}^2}^{k,l}(\boldsymbol{x}) + \bar{c}_{\mathbf{W}^1}^{k,l}(\boldsymbol{x}) \cdot c_{\mathbf{W}^2}^{k,l}(\boldsymbol{x})\right] \ge$$
$$\epsilon \cdot \min\left[\sum_{\boldsymbol{x}\in\mathcal{U}} c_{\mathbf{W}^1}^{k,l}(\boldsymbol{x}) \cdot c_{\mathbf{W}^2}^{k,l}(\boldsymbol{x}), \sum_{\boldsymbol{x}\in\mathcal{U}} \bar{c}_{\mathbf{W}^1}^{k,l}(\boldsymbol{x}) \cdot \bar{c}_{\mathbf{W}^2}^{k,l}(\boldsymbol{x})\right] \tag{9}$$

Here, $\bar{c}_{\mathbf{W}}^{k,l}(\boldsymbol{x}) = 1 - c_{\mathbf{W}}^{k,l}(\boldsymbol{x})$ indicates that the classification model $\mathbf{W}$ is not confident on its ranking prediction on $\boldsymbol{x}$. Conceptually speaking, the LHS of Eq.(9) corresponds to cases where exactly one of the two classification models $\mathbf{W}_1$ and $\mathbf{W}_2$ has confident ranking prediction, while the two terms in the min operator of RHS corresponds to cases where both classification models have or do not have confident ranking prediction.

By combing Eqs.(5), (6) and (9), Coins generates classification models by solving the following optimization problem:

$$\min_{\{\mathbf{W}^1,\mathbf{W}^2\}} \log\left(e^{V(\mathbf{W}^1,\mathcal{I}^1)}+e^{V(\mathbf{W}^2,\mathcal{I}^2)}\right)$$
$$\text{s.t.}: \sum_{a=1}^{d} \left(w_{ka}^1\right)^2 \cdot \left(w_{ka}^2\right)^2 = 0 \ (\forall\ 1 \le k \le q)$$
$$\sum_{\boldsymbol{x}\in\mathcal{U}} \left[c_{\mathbf{W}^1}^{k,l}(\boldsymbol{x}) \cdot \bar{c}_{\mathbf{W}^2}^{k,l}(\boldsymbol{x}) + \bar{c}_{\mathbf{W}^1}^{k,l}(\boldsymbol{x}) \cdot c_{\mathbf{W}^2}^{k,l}(\boldsymbol{x})\right] \ge$$
$$\epsilon \cdot \min\left[\sum_{\boldsymbol{x}\in\mathcal{U}} c_{\mathbf{W}^1}^{k,l}(\boldsymbol{x}) \cdot c_{\mathbf{W}^2}^{k,l}(\boldsymbol{x}), \sum_{\boldsymbol{x}\in\mathcal{U}} \bar{c}_{\mathbf{W}^1}^{k,l}(\boldsymbol{x}) \cdot \bar{c}_{\mathbf{W}^2}^{k,l}(\boldsymbol{x})\right]$$
$$(\forall\ 1 \le k \ne l \le q) \tag{10}$$

In this paper, the constrained optimization problem of Eq.(10) is solved by applying the augmented Lagrangian method [3, 8].

3.2 Supervision Information Communication

Following the iterative refinement strategy of co-training, Coins utilizes current classification models' predictions on unlabeled data to enrich the supervision information for classifier update. Other than binary predictions on the class labels, Coins chooses to utilize *pairwise ranking predictions* as the supervision information to be communicated between the classification models.

Specifically, let $\mathcal{J} \subseteq \{L+1, \ldots, L+U\}$ denote the index set of unlabeled data available at current co-training round. For each unlabeled data $\boldsymbol{x}_j$ ($j \in \mathcal{J}$), the empirical ranking loss of $\mathbf{W}^1$ on $\boldsymbol{x}_j$, i.e. $RL(\mathbf{W}^1, \mathcal{I}_j^1)$, is measured according to index set $\mathcal{I}_j^1$:

$$\mathcal{I}_j^1 = \{(j,k,l) \mid \langle \boldsymbol{w}_k^1, \boldsymbol{x}_j\rangle > 0, \langle \boldsymbol{w}_l^1, \boldsymbol{x}_j\rangle < 0, 1 \le k \ne l \le q\} \tag{11}$$

To obtain the enriching supervision information for updating the other classification model $\mathbf{W}^2$, Coins forms a ranking index set Δ^1 by identifying n unlabeled data which have *least* empirical ranking loss $RL(\mathbf{W}^1, \mathcal{I}_j^1)$. For each of the identified unlabeled data, one element from $\mathcal{I}_j^1$ is randomly picked up and added to Δ^1. Thereafter, the supervision information conveyed by Δ^1 is communicated to $\mathbf{W}^2$ for model update.[2] Equivalently, the same procedure can be invoked to obtain the enriching supervision information for updating $\mathbf{W}^1$.

Table 1 summarizes the pseudo-code of Coins.[3] Firstly, Coins initializes the ranking index sets from the labeled data set $\mathcal{L}$ (Step 1) and replenishes the pool of unlabeled data with unlabeled data

[2] Other than picking up all elements from $\mathcal{I}_j^1$ to form Δ^1, only one element is randomly picked up by Coins to avoid overfitting in supervision information communication.

[3] Code package for Coins is publicly-available at: http://cse.seu.edu.cn/PersonalPage/zhangml/Resources.htm#kdd17

Table 1: The pseudo-code of Coins.

Inputs:

$\mathcal{L}$: the set of L labeled data $\{(\boldsymbol{x}_i, \boldsymbol{y}_i) \mid 1 \le i \le L\}$

$\mathcal{U}$: the set of U unlabeled data $\{\boldsymbol{x}_i \mid L+1 \le i \le L+U\}$

C: the balancing parameter

ϵ: the ϵ-expansion parameter

Outputs:

$\mathbf{W}^1, \mathbf{W}^2$: the classification models

Process:

1: Initialize $\mathcal{I}^1$ and $\mathcal{I}^2$ according to Eq.(3);
2: Initialize $\mathcal{J} = \{L+1, \ldots, L+U\}$;
3: **repeat**
4: Optimize Eq.(10) with augmented Lagrangian method to obtain $\mathbf{W}^1$ and $\mathbf{W}^2$;
5: **for** $z \in \{1, 2\}$ **do**
6: Compute the empirical ranking loss $RL(\mathbf{W}^z, \mathcal{I}_j^z)$ $(j \in \mathcal{J})$;
7: Identify n unlabeled data with least empirical ranking loss $RL(\mathbf{W}^z, \mathcal{I}_j^z)$;
8: Form ranking index set Δ^z by randomly picking up one element from $\mathcal{I}_j^z$;
9: **end for**
10: Communicate Δ^1 to $\mathbf{W}^2$ by updating $\mathcal{I}^2 = \mathcal{I}^2 \bigcup \Delta^1$;
11: Communicate Δ^2 to $\mathbf{W}^1$ by updating $\mathcal{I}^1 = \mathcal{I}^1 \bigcup \Delta^2$;
12: **for** $(j, k, l) \in \Delta^1 \bigcup \Delta^2$ **do**
13: $\mathcal{U} = \mathcal{U} \setminus \{\boldsymbol{x}_j\}$;
14: $\mathcal{J} = \mathcal{J} \setminus \{j\}$;
15: **end for**
16: **until** convergence
17: Return $\mathbf{W}^1$ and $\mathbf{W}^2$.

set $\mathcal{U}$ (Step 2). Then, in each co-training round Coins alternates between two phases until convergence: a) Generate two classification models based on current ranking index sets and pool of unlabeled data (Step 4); b) Communicate supervision information between two classification models by updating ranking index sets and pool of unlabeled data (Steps 5-15). Finally, two classification models are returned whose modeling outputs are averaged to make prediction for unseen instance.[4]

4 EXPERIMENTS

4.1 Experimental Setup

4.1.1 Data Sets. To thoroughly evaluate the performance of the proposed approach, a total of ten benchmark multi-label data sets have been employed for experimental studies. Given the multi-label data set $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) \mid 1 \le i \le N\}$, we use $|\mathcal{D}|$, $dim(\mathcal{D})$, $CL(\mathcal{D})$ to represent its number of *examples*, *features*, *class labels* and $F(\mathcal{D})$ to represent its *feature type*. Furthermore, several other properties of multi-label data sets are denoted as [29]:

- *Label cardinality*: $LCard(\mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{q}\mathbb{I}(y_{ik} = +1)$ which counts the average number of relevant labels per example;
- *Label density*: $LDen(\mathcal{D}) = \frac{LCard(\mathcal{D})}{CL(\mathcal{D})}$ which normalizes label cardinality by the number of class labels;
- *Distinct label sets*: $DL(\mathcal{D}) = |\{\boldsymbol{y} \mid \exists \boldsymbol{x} : (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}\}|$ which counts the number of distinct label vectors (relevant label set) existing in $\mathcal{D}$;
- *Proportion of distinct label sets*: $PDL(\mathcal{D}) = \frac{DL(\mathcal{D})}{|\mathcal{D}|}$ which normalizes $DL(\mathcal{D})$ by the number of examples.

Table 2 summarizes the detailed characteristics of the experimental data sets.[5] As shown in Table 2, the experimental data sets have diversified multi-label properties which serve as a solid basis for comprehensive comparative studies. To the best of our knowledge, in most cases existing works on semi-supervised multi-label learning [7, 14, 15, 17, 21, 22, 28] only employed data sets with less than 5000 examples in their experimental studies.

4.1.2 Comparing Algorithms. In this paper, the performance of Coins is compared against four state-of-the-art algorithms, including one *fully supervised*, two *transductive semi-supervised*, and one *inductive semi-supervised* multi-label learning approaches:

- Ecc [20]: Ecc works in a fully supervised manner by transforming the multi-label learning problem into a chain of binary classification problems, where predictions of preceding classifiers are used as extra features to learn subsequent classifiers in the chain. For Ecc, ensemble learning is employed to exploit the randomness of chaining order and is shown to achieve highly competitive performance in learning from multi-label data [12, 29];
- Smse [7]: Smse works in a transductive semi-supervised manner by enforcing manifold regularization on the labeled and unlabeled data, where the resulting optimization problem is equivalent to solve a Sylvester equation. For Smse, the affinity matrix used to enable manifold regularization is instantiated based on a fully-connected and symmetric graph over labeled and unlabeled data;
- Tram [17]: Tram works in a transductive semi-supervised manner by conducting label propagation on the labeled and unlabeled data, where the resulting optimization problem is equivalent to solve a system of linear equations with closed-form solution. For Tram, the affinity matrix used to enable label propagation is instantiated based on a sparse and asymmetric kNN graph over labeled and unlabeled data;
- iMlcu [26]: iMlcu works in an inductive semi-supervised manner by imposing large margin criterion on the labeled and unlabeled data, where the resulting optimization problem is non-convex and admits an iterative CCCP solution. For iMlcu, empirical ranking loss and pseudo hinge loss are used to instantiate the objective terms on labeled and unlabeled data respectively.

[4] In this paper, the number of unlabeled data identified in Step 7 (i.e. n) is set to be $\max\left(\frac{1}{10} \cdot L, 5\right)$. Furthermore, the iterative co-training procedure terminates when the pool of unlabeled data is empty or the number of co-training rounds reaches 30.

[5] Publicly available at http://mulan.sourceforge.net/datasets.html and http://meka.sourceforge.net/#datasets

Table 2: Characteristics of the benchmark multi-label data sets.

Data set	$\|\mathcal{D}\|$	$dim(\mathcal{D})$	$CL(\mathcal{D})$	$F(\mathcal{D})$	$LCard(\mathcal{D})$	$LDen(\mathcal{D})$	$DL(\mathcal{D})$	$PDL(\mathcal{D})$	Domain
enron	1702	1001	16	nominal	2.854	0.178	356	0.209	text
image	2000	294	5	numeric	1.236	0.247	20	0.010	images
scene	2407	294	6	numeric	1.074	0.179	15	0.006	images
yeast	2417	103	14	numeric	4.237	0.303	198	0.082	biology
slashdot	3782	1079	14	nominal	1.135	0.081	120	0.032	text
corel5k	5000	499	38	nominal	2.090	0.055	894	0.179	images
rcv1-subset1	6000	472	30	numeric	2.171	0.072	379	0.063	text
arts1	7484	231	17	numeric	1.585	0.093	478	0.064	text
eurlex-dc	19348	100	41	numeric	0.703	0.017	182	0.009	text
eurlex-sm	19348	100	20	numeric	1.337	0.067	352	0.018	text

4.1.3 Evaluation Protocol. For each data set in Table 2, we randomly sample $\alpha \times 100\%$ examples to form the labeled data set $\mathcal{L}$. For the remaining examples, 40% of them are randomly sampled to form the unlabeled data set $\mathcal{U}$ and 10% of them are randomly sampled to form the test data set $\mathcal{T}$. In the following experiments, the training and testing procedures are conducted for the comparing algorithms as follows:

- For fully supervised learning algorithm Ecc, it is trained on the labeled data set $\mathcal{L}$ and evaluated on the test data set $\mathcal{T}$;
- For inductive semi-supervised learning algorithms Coins and iMlcu, both of them are trained on $\mathcal{L} \bigcup \mathcal{U}$ and evaluated on $\mathcal{T}$;
- For transductive semi-supervised learning algorithms Smse and Tram, both of them can only make predictions on unlabeled data which have been used in the training phase. To ensure that all comparing algorithms are evaluated on the same test data set, Smse and Tram are trained on $\mathcal{L} \bigcup \mathcal{U} \bigcup \mathcal{T}$ and evaluated in $\mathcal{T}$. Therefore, Smse and Tram have access to more unlabeled data (i.e. $\mathcal{U} \bigcup \mathcal{T}$) during the training phase than Coins and iMlcu (i.e. $\mathcal{U}$).

In this paper, we vary the sampling rate α for labeled data from 1% to 5% with a step-size of 1%. Under each sampling rate, ten times of random sampling are performed on each data set, where the mean performance as well as standard deviation out of ten runs of experiments are recorded for the comparing algorithms. Accordingly, six widely-used multi-label metrics are employed for performance evaluation including *hamming loss*, *one-error*, *coverage*, *ranking loss*, *average precision*, and *macro-averaging AUC*. These metrics evaluate the performance of multi-label predictive models from various aspects, whose detailed definitions can be found in [29]. For *hamming loss*, *one-error*, *coverage* and *ranking loss*, the smaller the metric values the better the performance. For *average precision* and *macro-averaging AUC*, the greater the metric values the better the performance.

As shown in Table 1, the balancing parameter C for Coins is set via cross-validation and the ϵ-expansion parameter is set to be 10 across all data sets. In addition, parameter configurations suggested in respective literatures are used for the other comparing algorithms [7, 17, 20, 26].

4.2 Experimental Results

Tables 3, 4 and 5 report the detailed experimental results of each comparing algorithm on all data sets in terms of *hamming loss*, *ranking loss* and *macro-averaging AUC* respectively.[6] By fixing the sampling rate α for labeled data as well as the evaluation metric, performance comparisons are carried out among multiple algorithms over a number of data sets. Consequently, the favorable *Friedman test* [10] is utilized in this paper to systematically analyze the relative performance among the comparing algorithms.

Table 6 summarizes the Friedman statistics F_F and the corresponding critical value across each sampling rate α and evaluation metric. As shown in Table 6 (inductive setting), at 0.05 significance level, the null hypothesis of indistinguishable performance among the comparing algorithms should be rejected under each configuration. Thereafter, the *Bonferroni-Dunn test* [10] is employed as the post-hoc test to elicit the relative performance, where Coins is treated as the control algorithm in the statistical test. The average rank over all data sets is recorded for each algorithm, where the difference between Coins and one comparing algorithm is calibrated with the *critical difference* (CD). Here, the performance difference is deemed to be significant if their average ranks differ by at least one CD (CD=1.7664 in this paper; # comparing algorithms k=5, # data sets N=10).

To visually manifest the relative performance among the comparing algorithms, Figure 1 illustrates the CD diagram [10] on each sampling rate α for labeled data in terms of *hamming loss*, *ranking loss* and *macro-averaging AUC*. Similarly, CD diagrams in terms of the other evaluation metrics have also been made available at the aforementioned link in footnote 6. In each CD diagram, the average rank of comparing algorithms is marked along the axis with lower ranks to the right. Furthermore, any comparing algorithm with average rank within one CD to that of Coins is interconnected to each other with a thick line. Otherwise, its performance is deemed to be significantly different to Coins.

Accordingly, the following observations can be made based on the experimental results:

[6] Due to page limit, detailed experimental results in terms of the other evaluation metrics have been made available at: http://cse.seu.edu.cn/PersonalPage/zhangml/files/Coins_FullResults.pdf

Table 3: Inductive performance of each comparing algorithm (mean ± std. deviation) in terms of *hamming Loss*, where the sampling rate α for labeled data varies from 1% to 5% with step-size of 1%.

Comparing algorithm	α = 1%									
	enron	image	scene	yeast	slashdot	corel5k	rcv1-subset1	arts1	eurlex-dc	eurlex-sm
Coins	0.178±0.014	0.273±0.022	0.172±0.019	0.245±0.013	0.081±0.002	0.055±0.002	0.073±0.003	0.094±0.002	0.018±0.001	0.066±0.001
Ecc	0.183±0.011	0.288±0.016	0.164±0.018	0.254±0.014	0.090±0.007	0.071±0.004	0.095±0.004	0.131±0.019	0.022±0.001	0.070±0.002
Smse	0.430±0.036	0.309±0.052	0.281±0.012	0.239±0.018	0.160±0.067	0.122±0.071	0.072±0.001	0.092±0.001	0.017±0.000	0.066±0.001
Tram	0.205±0.022	0.313±0.019	0.182±0.025	0.257±0.010	0.110±0.016	0.090±0.009	0.107±0.007	0.138±0.010	0.018±0.003	0.068±0.005
iMlcu	0.161±0.010	0.285±0.020	0.271±0.035	0.255±0.014	0.080±0.003	0.058±0.002	0.073±0.003	0.092±0.001	0.017±0.000	0.066±0.001
Comparing algorithm	α = 2%									
	enron	image	scene	yeast	slashdot	corel5k	rcv1-subset1	arts1	eurlex-dc	eurlex-sm
Coins	0.164±0.012	0.229±0.009	0.149±0.008	0.236±0.016	0.079±0.003	0.055±0.001	0.071±0.001	0.094±0.002	0.018±0.000	0.067±0.001
Ecc	0.170±0.010	0.249±0.015	0.146±0.012	0.237±0.010	0.089±0.005	0.083±0.003	0.088±0.003	0.141±0.003	0.020±0.001	0.065±0.002
Smse	0.458±0.038	0.254±0.022	0.186±0.006	0.236±0.013	0.152±0.063	0.196±0.059	0.072±0.001	0.093±0.001	0.017±0.000	0.067±0.001
Tram	0.193±0.011	0.254±0.026	0.150±0.015	0.240±0.014	0.111±0.003	0.086±0.006	0.100±0.003	0.137±0.005	0.021±0.005	0.065±0.002
iMlcu	0.157±0.009	0.257±0.011	0.207±0.011	0.237±0.016	0.077±0.003	0.061±0.002	0.073±0.002	0.093±0.001	0.017±0.000	0.067±0.001
Comparing algorithm	α = 3%									
	enron	image	scene	yeast	slashdot	corel5k	rcv1-subset1	arts1	eurlex-dc	eurlex-sm
Coins	0.152±0.007	0.228±0.015	0.133±0.009	0.239±0.010	0.078±0.005	0.056±0.001	0.071±0.001	0.093±0.002	0.018±0.000	0.067±0.001
Ecc	0.164±0.016	0.228±0.019	0.134±0.016	0.234±0.013	0.094±0.003	0.086±0.002	0.086±0.002	0.141±0.003	0.019±0.000	0.062±0.002
Smse	0.411±0.079	0.250±0.017	0.236±0.045	0.237±0.012	0.120±0.064	0.180±0.112	0.073±0.001	0.093±0.002	0.017±0.000	0.067±0.001
Tram	0.181±0.014	0.248±0.019	0.122±0.010	0.231±0.006	0.106±0.005	0.086±0.003	0.095±0.002	0.136±0.007	0.017±0.003	0.065±0.004
iMlcu	0.157±0.013	0.235±0.019	0.180±0.022	0.237±0.014	0.089±0.015	0.064±0.002	0.076±0.002	0.092±0.002	0.017±0.000	0.067±0.001
Comparing algorithm	α = 4%									
	enron	image	scene	yeast	slashdot	corel5k	rcv1-subset1	arts1	eurlex-dc	eurlex-sm
Coins	0.148±0.015	0.220±0.013	0.136±0.007	0.231±0.011	0.074±0.002	0.057±0.001	0.070±0.001	0.092±0.002	0.017±0.001	0.067±0.001
Ecc	0.159±0.012	0.224±0.018	0.128±0.006	0.229±0.011	0.097±0.003	0.086±0.002	0.085±0.002	0.141±0.007	0.019±0.001	0.060±0.001
Smse	0.448±0.036	0.234±0.006	0.272±0.023	0.227±0.007	0.120±0.062	0.135±0.057	0.073±0.001	0.092±0.002	0.017±0.000	0.067±0.001
Tram	0.175±0.016	0.235±0.013	0.120±0.010	0.225±0.009	0.101±0.003	0.086±0.004	0.087±0.003	0.135±0.007	0.018±0.003	0.062±0.003
iMlcu	0.152±0.011	0.220±0.016	0.185±0.014	0.226±0.008	0.081±0.004	0.070±0.002	0.079±0.003	0.092±0.002	0.017±0.000	0.066±0.001
Comparing algorithm	α = 5%									
	enron	image	scene	yeast	slashdot	corel5k	rcv1-subset1	arts1	eurlex-dc	eurlex-sm
Coins	0.143±0.010	0.216±0.007	0.133±0.008	0.221±0.008	0.072±0.003	0.057±0.002	0.070±0.002	0.092±0.002	0.017±0.000	0.066±0.001
Ecc	0.155±0.006	0.228±0.010	0.122±0.008	0.223±0.010	0.097±0.005	0.086±0.002	0.082±0.003	0.137±0.005	0.018±0.001	0.059±0.002
Smse	0.453±0.033	0.244±0.016	0.229±0.049	0.224±0.007	0.185±0.056	0.167±0.092	0.072±0.002	0.092±0.002	0.017±0.001	0.066±0.001
Tram	0.167±0.008	0.233±0.017	0.122±0.015	0.223±0.008	0.099±0.005	0.085±0.003	0.082±0.003	0.134±0.004	0.017±0.002	0.063±0.003
iMlcu	0.147±0.007	0.221±0.011	0.184±0.015	0.222±0.015	0.081±0.011	0.073±0.005	0.075±0.005	0.091±0.001	0.017±0.000	0.066±0.001

- Across all sampling rate α and evaluation metrics, it is impressive that Coins achieves optimal (lowest) average rank in 70% cases and the second best average rank in 26.7% cases. No algorithm has achieved significantly better performance than Coins in all cases.
- As shown in Table 1 (steps 5-9), empirical ranking loss has been utilized by Coins for controlling supervision information communication during each co-training round. Accordingly, Coins achieves optimal average rank in terms of ranking loss and significantly outperforms Ecc and Smse on all sampling rate α.
- Comparing to fully supervised and transductive semi-supervised multi-label algorithms, Coins achieves better average rank than Ecc, Smse, and Tram in 83.3%, 100%, and 96.7% cases respectively.
- Comparing to another inductive semi-supervised multi-label learning algorithm, Coins achieves better average rank than iMlcu in 76.7% cases and is more computationally efficient with less training time at an order of magnitude.

In addition to the inductive performance, transductive performance of comparing algorithm can also be evaluated by their predictive performance on the unlabeled data set $\mathcal{U}$. As shown in Table 6 (transductive setting), at 0.05 significance level, the null hypothesis of indistinguishable performance among the comparing algorithms should be rejected under each configuration.

Accordingly, Figure 2 illustrates the CD diagram in terms of *hamming loss*, *ranking loss* and *macro-averaging AUC* under the transductive setting. The following observations can be made on the transductive experimental results:

- Across all sampling rate α and evaluation metrics, Coins achieves optimal average rank in 66.7% cases and the second

Table 4: Inductive performance of each comparing algorithm (mean ± std. deviation) in terms of *ranking loss*, where the sampling rate α for labeled data varies from 1% to 5% with step-size of 1%.

Comparing algorithm	enron	image	scene	yeast	slashdot	corel5k	rcv1-subset1	arts1	eurlex-dc	eurlex-sm
	α = 1%									
Coins	0.233±0.017	0.336±0.037	0.201±0.042	0.226±0.012	0.315±0.018	0.335±0.020	0.211±0.021	0.260±0.009	0.284±0.012	0.295±0.017
Ecc	0.373±0.054	0.342±0.026	0.246±0.036	0.247±0.011	0.794±0.054	0.892±0.032	0.616±0.035	0.707±0.167	0.564±0.024	0.429±0.024
Smse	0.496±0.024	0.461±0.044	0.376±0.038	0.230±0.026	0.519±0.031	0.546±0.068	0.466±0.052	0.266±0.010	0.318±0.012	0.340±0.008
Tram	0.267±0.036	0.374±0.024	0.229±0.048	0.217±0.009	0.483±0.027	0.342±0.021	0.290±0.024	0.264±0.011	0.833±0.233	0.681±0.324
iMlcu	0.208±0.018	0.328±0.033	0.193±0.033	0.263±0.019	0.294±0.020	0.359±0.015	0.305±0.065	0.376±0.019	0.222±0.024	0.215±0.022
Comparing algorithm	enron	image	scene	yeast	slashdot	corel5k	rcv1-subset1	arts1	eurlex-dc	eurlex-sm
	α = 2%									
Coins	0.197±0.028	0.277±0.035	0.141±0.012	0.211±0.015	0.271±0.026	0.317±0.016	0.169±0.011	0.237±0.005	0.266±0.008	0.290±0.012
Ecc	0.359±0.053	0.291±0.032	0.190±0.025	0.226±0.020	0.726±0.026	0.816±0.031	0.476±0.025	0.586±0.056	0.520±0.015	0.404±0.024
Smse	0.491±0.031	0.414±0.061	0.337±0.049	0.222±0.021	0.542±0.026	0.503±0.054	0.407±0.049	0.259±0.006	0.301±0.005	0.340±0.006
Tram	0.219±0.014	0.295±0.039	0.160±0.044	0.202±0.013	0.293±0.015	0.299±0.015	0.224±0.011	0.247±0.006	0.588±0.381	0.865±0.102
iMlcu	0.187±0.008	0.284±0.027	0.151±0.017	0.234±0.020	0.264±0.017	0.361±0.015	0.247±0.017	0.338±0.027	0.195±0.014	0.200±0.008
Comparing algorithm	enron	image	scene	yeast	slashdot	corel5k	rcv1-subset1	arts1	eurlex-dc	eurlex-sm
	α = 3%									
Coins	0.185±0.019	0.250±0.029	0.120±0.013	0.215±0.011	0.239±0.024	0.304±0.018	0.148±0.008	0.229±0.005	0.261±0.012	0.295±0.008
Ecc	0.347±0.028	0.258±0.029	0.163±0.023	0.229±0.012	0.646±0.038	0.717±0.022	0.381±0.032	0.505±0.050	0.487±0.026	0.404±0.013
Smse	0.447±0.163	0.384±0.041	0.308±0.024	0.220±0.010	0.535±0.016	0.525±0.044	0.391±0.033	0.254±0.007	0.299±0.011	0.341±0.012
Tram	0.214±0.015	0.272±0.031	0.115±0.010	0.197±0.009	0.279±0.015	0.277±0.012	0.182±0.008	0.240±0.006	0.730±0.291	0.649±0.303
iMlcu	0.200±0.021	0.255±0.029	0.134±0.016	0.236±0.011	0.264±0.029	0.393±0.015	0.194±0.019	0.332±0.013	0.178±0.012	0.184±0.010
Comparing algorithm	enron	image	scene	yeast	slashdot	corel5k	rcv1-subset1	arts1	eurlex-dc	eurlex-sm
	α = 4%									
Coins	0.170±0.025	0.247±0.030	0.118±0.011	0.205±0.015	0.221±0.008	0.296±0.016	0.138±0.006	0.223±0.007	0.256±0.010	0.293±0.010
Ecc	0.322±0.031	0.250±0.025	0.149±0.019	0.219±0.016	0.583±0.033	0.671±0.015	0.338±0.015	0.512±0.053	0.476±0.014	0.388±0.014
Smse	0.501±0.021	0.340±0.030	0.303±0.031	0.212±0.012	0.524±0.030	0.528±0.061	0.367±0.036	0.250±0.007	0.296±0.011	0.339±0.005
Tram	0.198±0.014	0.254±0.026	0.112±0.009	0.193±0.012	0.252±0.013	0.265±0.015	0.147±0.003	0.237±0.011	0.669±0.313	0.767±0.119
iMlcu	0.187±0.017	0.254±0.027	0.127±0.011	0.228±0.016	0.244±0.010	0.379±0.014	0.171±0.011	0.325±0.014	0.167±0.015	0.173±0.008
Comparing algorithm	enron	image	scene	yeast	slashdot	corel5k	rcv1-subset1	arts1	eurlex-dc	eurlex-sm
	α = 5%									
Coins	0.158±0.011	0.242±0.016	0.119±0.019	0.195±0.012	0.214±0.014	0.295±0.010	0.129±0.010	0.217±0.013	0.255±0.010	0.291±0.011
Ecc	0.313±0.022	0.253±0.017	0.148±0.019	0.216±0.014	0.550±0.024	0.622±0.024	0.295±0.022	0.423±0.066	0.460±0.021	0.390±0.014
Smse	0.474±0.034	0.333±0.040	0.294±0.018	0.208±0.012	0.502±0.029	0.530±0.043	0.355±0.044	0.247±0.007	0.291±0.010	0.340±0.007
Tram	0.178±0.011	0.251±0.017	0.111±0.013	0.189±0.009	0.251±0.018	0.252±0.013	0.124±0.008	0.227±0.006	0.735±0.258	0.843±0.098
iMlcu	0.181±0.013	0.259±0.021	0.136±0.019	0.223±0.017	0.241±0.030	0.370±0.013	0.168±0.010	0.316±0.011	0.156±0.014	0.173±0.010

best average rank in 26.7% cases. No algorithm has achieved significantly better performance than Coins in all cases.

- Coins achieves optimal average rank in terms of ranking loss and significantly outperforms Ecc and Smse on all sampling rate α. Furthermore, the average rank of Coins is lower than Ecc, Smse, Tram, and iMlcu in 83.3%, 100%, 96.7%, and 80.0% cases respectively.

To summarize, Coins performs favorably against the state-of-the-art comparing algorithms across diverse data sets, sampling rate of labeled data and evaluation metrics. These results clearly validate the effectiveness of employing co-training to facilitate semi-supervised multi-label learning with strong inductive as well as transductive predictive performance.

5 CONCLUSION

In this paper, the problem of inductive semi-supervised multi-label learning is addressed where a novel approach named Coins is proposed by adapting the well-known co-training strategy. To enable co-training for handling multi-label data, two classification models are generated by dichotomizing the feature space with diversity maximization, and then pairwise ranking predictions on unlabeled data is iteratively communicated for model refinement. Comprehensive experimental studies have been conducted to show the effectiveness of the proposed co-training based semi-supervised multi-label learning approach.

REFERENCES

[1] J. St. Amand and J. Huan. 2016. Discriminative view learning for single view co-training. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. Indianapolis, IN, 2221–2226.

Table 5: Inductive performance of each comparing algorithm (mean ± std. deviation) in terms of *macro-averaging AUC*, where the sampling rate α for labeled data varies from 1% to 5% with step-size of 1%.

Comparing algorithm	enron	image	scene	yeast	slashdot	corel5k	rcv1-subset1	arts1	eurlex-dc	eurlex-sm
	$\alpha = 1\%$									
Coins	0.549±0.040	0.669±0.027	0.825±0.027	0.562±0.039	0.622±0.038	0.568±0.014	0.734±0.029	0.559±0.015	0.686±0.024	0.697±0.015
Ecc	0.574±0.019	0.687±0.035	0.809±0.026	0.570±0.040	0.542±0.009	0.512±0.007	0.592±0.016	0.506±0.005	0.591±0.011	0.686±0.011
Smse	0.546±0.018	0.610±0.033	0.518±0.021	0.580±0.038	0.519±0.018	0.504±0.016	0.538±0.018	0.555±0.015	0.692±0.025	0.697±0.018
Tram	0.614±0.031	0.664±0.035	0.873±0.014	0.573±0.045	0.617±0.066	0.599±0.022	0.793±0.022	0.534±0.015	0.531±0.075	0.580±0.105
iMlcu	0.641±0.026	0.666±0.037	0.715±0.063	0.555±0.039	0.671±0.013	0.574±0.015	0.585±0.115	0.566±0.018	0.742±0.019	0.754±0.014
	$\alpha = 2\%$									
Coins	0.609±0.024	0.722±0.023	0.861±0.013	0.582±0.026	0.684±0.022	0.591±0.020	0.777±0.018	0.578±0.017	0.712±0.015	0.719±0.014
Ecc	0.604±0.018	0.735±0.024	0.855±0.010	0.590±0.016	0.560±0.005	0.521±0.009	0.642±0.018	0.515±0.009	0.613±0.009	0.701±0.009
Smse	0.548±0.012	0.670±0.032	0.547±0.014	0.601±0.015	0.544±0.014	0.506±0.011	0.554±0.012	0.573±0.013	0.726±0.010	0.713±0.014
Tram	0.636±0.015	0.725±0.029	0.884±0.016	0.603±0.030	0.677±0.024	0.630±0.017	0.842±0.012	0.556±0.015	0.636±0.141	0.530±0.029
iMlcu	0.684±0.018	0.698±0.018	0.810±0.021	0.581±0.019	0.703±0.016	0.588±0.018	0.634±0.026	0.583±0.012	0.783±0.012	0.772±0.009
	$\alpha = 3\%$									
Coins	0.646±0.024	0.746±0.026	0.878±0.012	0.588±0.024	0.722±0.022	0.614±0.014	0.807±0.010	0.592±0.016	0.723±0.010	0.724±0.016
Ecc	0.613±0.020	0.763±0.023	0.866±0.017	0.583±0.027	0.598±0.009	0.536±0.009	0.693±0.019	0.519±0.004	0.630±0.011	0.711±0.009
Smse	0.486±0.172	0.665±0.041	0.538±0.027	0.594±0.021	0.534±0.014	0.501±0.015	0.559±0.017	0.591±0.016	0.742±0.011	0.730±0.011
Tram	0.648±0.028	0.736±0.030	0.906±0.010	0.601±0.015	0.710±0.018	0.641±0.019	0.868±0.013	0.575±0.011	0.577±0.105	0.614±0.115
iMlcu	0.684±0.024	0.731±0.032	0.843±0.021	0.586±0.030	0.614±0.061	0.522±0.023	0.749±0.047	0.604±0.012	0.805±0.006	0.796±0.012
	$\alpha = 4\%$									
Coins	0.679±0.040	0.752±0.028	0.882±0.013	0.605±0.021	0.731±0.021	0.622±0.017	0.812±0.009	0.597±0.007	0.738±0.012	0.731±0.009
Ecc	0.626±0.018	0.769±0.023	0.878±0.020	0.602±0.020	0.618±0.010	0.548±0.011	0.714±0.014	0.523±0.009	0.640±0.010	0.718±0.006
Smse	0.530±0.022	0.700±0.024	0.551±0.016	0.626±0.027	0.550±0.019	0.505±0.013	0.558±0.009	0.592±0.009	0.763±0.009	0.733±0.008
Tram	0.671±0.029	0.751±0.029	0.907±0.012	0.618±0.020	0.719±0.017	0.659±0.025	0.876±0.007	0.579±0.012	0.602±0.129	0.588±0.053
iMlcu	0.693±0.026	0.749±0.026	0.848±0.021	0.602±0.022	0.636±0.055	0.564±0.018	0.787±0.027	0.608±0.013	0.820±0.007	0.806±0.013
	$\alpha = 5\%$									
Coins	0.692±0.026	0.751±0.016	0.885±0.014	0.612±0.012	0.739±0.022	0.628±0.011	0.823±0.012	0.603±0.009	0.744±0.015	0.737±0.015
Ecc	0.635±0.020	0.768±0.016	0.883±0.013	0.611±0.011	0.630±0.008	0.560±0.007	0.743±0.016	0.534±0.010	0.650±0.009	0.721±0.005
Smse	0.546±0.017	0.711±0.030	0.539±0.016	0.630±0.020	0.547±0.016	0.503±0.012	0.559±0.012	0.598±0.010	0.769±0.015	0.738±0.010
Tram	0.685±0.033	0.754±0.014	0.907±0.011	0.625±0.019	0.739±0.011	0.676±0.016	0.887±0.007	0.590±0.012	0.583±0.110	0.563±0.041
iMlcu	0.701±0.025	0.742±0.022	0.842±0.024	0.605±0.012	0.655±0.040	0.607±0.019	0.807±0.010	0.617±0.013	0.827±0.008	0.810±0.009

Table 6: Summary of the Friedman statistics F_F and the critical value (at 0.05 significance level) across each sampling rate α and evaluation metric (# comparing algorithms $k = 5$, # data sets $N = 10$).

Evaluation metric	F_F										critical value
	inductive setting					transductive setting					
	$\alpha = 1\%$	$\alpha = 2\%$	$\alpha = 3\%$	$\alpha = 4\%$	$\alpha = 5\%$	$\alpha = 1\%$	$\alpha = 2\%$	$\alpha = 3\%$	$\alpha = 4\%$	$\alpha = 5\%$	
hamming loss	14.8000	7.6000	6.3200	7.2000	13.0400	12.9600	10.2400	7.0400	7.6000	11.1200	2.6335
one-error	17.3600	16.3200	15.7600	15.7600	19.2000	16.4000	15.9200	15.7600	16.0800	16.8000	
coverage	11.2000	10.5600	12.4800	11.6000	11.8400	10.5600	10.9600	12.6400	11.4400	12.8000	
ranking loss	20.0000	19.8400	19.7600	18.6400	19.6000	20.9600	19.8400	18.6400	18.5600	17.8400	
average precision	11.8400	15.2800	16.4000	15.4400	16.7200	16.2400	14.9600	16.9600	18.4800	15.7600	
macro-averaging AUC	6.1400	11.2000	10.9600	8.5400	7.6000	7.4400	7.7600	8.6400	6.3200	6.8800	

[2] M.-F. Balcan, A. Blum, and K. Yang. 2004. Co-training and expansion: Towards bridging theory and practice. In *Advances in Neural Information Processing Systems 17*, L. Saul, Y. Weiss, and L. Bottou (Eds.). MIT Press, Cambridge, MA, 89–96.

[3] D. P. Bertsekas. 2016. *Nonlinear Programming* (3rd ed.). Athena Scientific, Belmont, MA.

[4] A. Blum and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning*

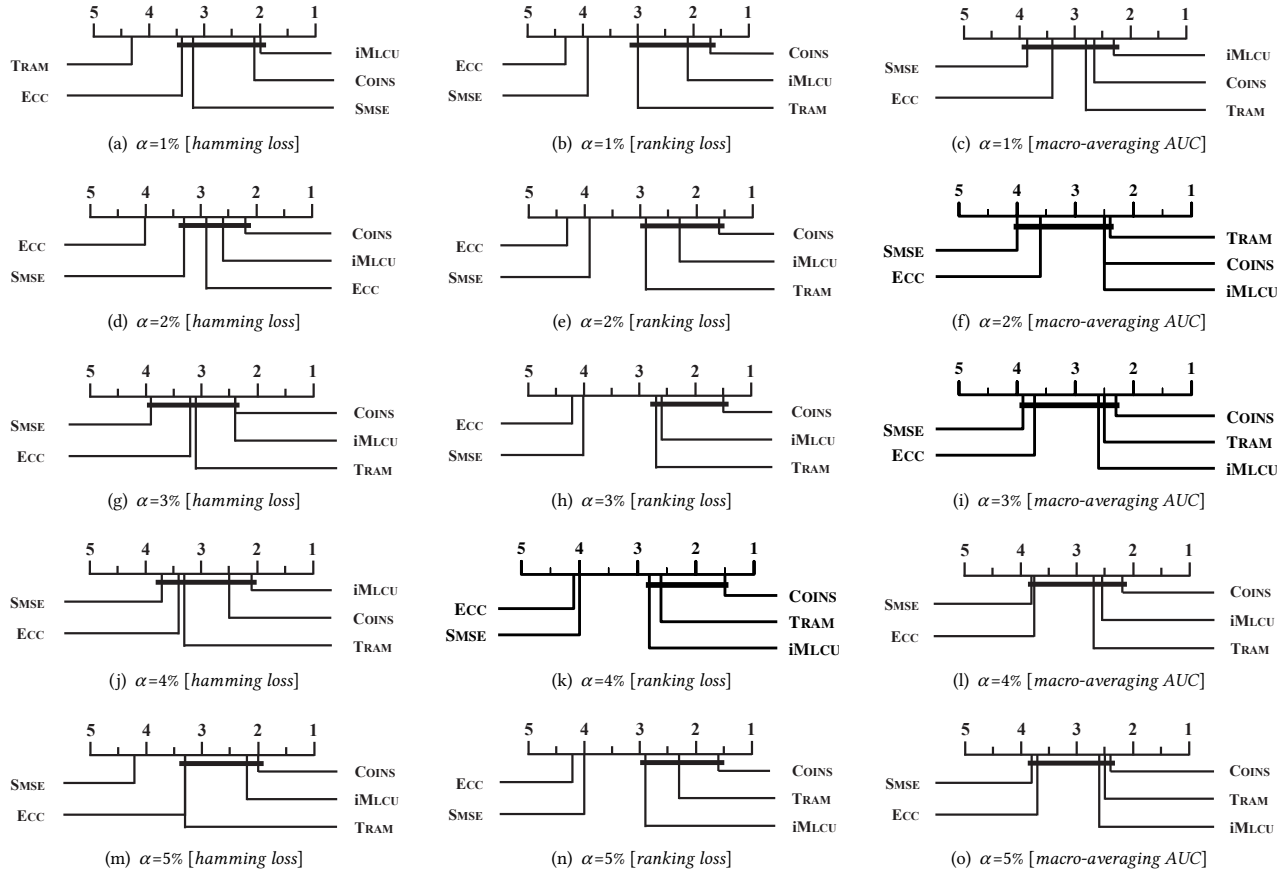

Figure 1: Comparison of COINS (control algorithm) against other comparing algorithms with the *Bonferroni-Dunn test* (inductive setting). Algorithms not connected with COINS in the CD diagram are considered to have significantly different performance from the control algorithm (CD=1.7664 at 0.05 significance level). Left column: *hamming loss*; Middle column: *ranking loss*; Right column: *macro-averaging AUC*.

Theory. Madison, WI, 92–100.

[5] X. Chang, F. Nie, Y. Yang, and H. Huang. 2014. A convex formulation for semi-supervised multi-label feature selection. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*. Quebec City, Canada, 1171–1177.

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

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

[8] M. Chen, K. Q. Weinberger, and Y. Chen. 2011. Automatic feature decomposition for single view co-training. In *Proceedings of 28th International Conference on Machine Learning*. Bellevue, WA, 953–960.

[9] R. Collobert, F. Sinz, J. Weston, and L. Bottou. 2006. Large scale transductive SVMs. *Journal of Machine Learning Research* 7 (2006), 1687–1712.

[10] J. Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7, Jan (2006), 1–30.

[11] J. Du, C. X. Ling, and Z.-H. Zhou. 2011. When does co-training work in real data? *IEEE Transactions on Knowledge and Data Engineering* 23, 5 (2011), 788–799.

[12] E. Gibaja and S. Ventura. 2015. A tutorial on multilabel learning. *Comput. Surveys* 47, 3 (2015), Article 52.

[13] S. Goldman and Y. Zhou. 2000. Enhancing supervised learning with unlabeled data. In *Proceedings of 17th International Conference on Machine Learning*. San Francisco, CA, 327–334.

[14] Y.-H. Guo and D. Schuurmans. 2012. Semi-supervised multi-Label classification: A simultaneous large-margin, subspace learning approach. In *Lecture Notes in Computer Science 7524*, P.-A. Flach, T.-D. Bie, and N. Cristianini (Eds.). Springer, Berlin, 355–370.

[15] L. Jing, L. Yang, J. Yu, and M. K. Ng. 2015. Semi-supervised low-rank mapping learning for multi-label classification. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*. Boston, MA, 1483–1491.

[16] T. Joachims. 1999. Transductive inference for text classification using support vector machines. In *Proceedings of 16th International Conference on Machine Learning*. Bled, Slovenia, 200–209.

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

[18] A. C. E. S. Lima and L. N. de Castro. 2014. A multi-label, semi-supervised classification approach applied to personality prediction in social media. *Neural Networks* 58 (2014), 122–130.

[19] Y. Luo, D. Tao, B. Geng, C. Xu, and S. J. Maybank. 2013. Manifold regularized multitask learning for semi-supervised multilabel image classification. *IEEE Transactions on Image Processing* 22, 2 (2013), 523–536.

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

[21] B. Wang and J. Tsotsos. 2016. Dynamic label propagation for semi-supervised multi-class multi-label classification. *Pattern Recognition* 52 (2016), 75–84.

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

[23] W. Wang and Z.-H. Zhou. 2007. Analyzing co-training style algorithms. In *Lecture Notes in Artificial Intelligence 4701*, J. N. Kok, J. Koronacki, R. L. Mántaras, S. Matwin, D. Mladenic, and A. Skowron (Eds.). Springer, Berlin, 454–465.

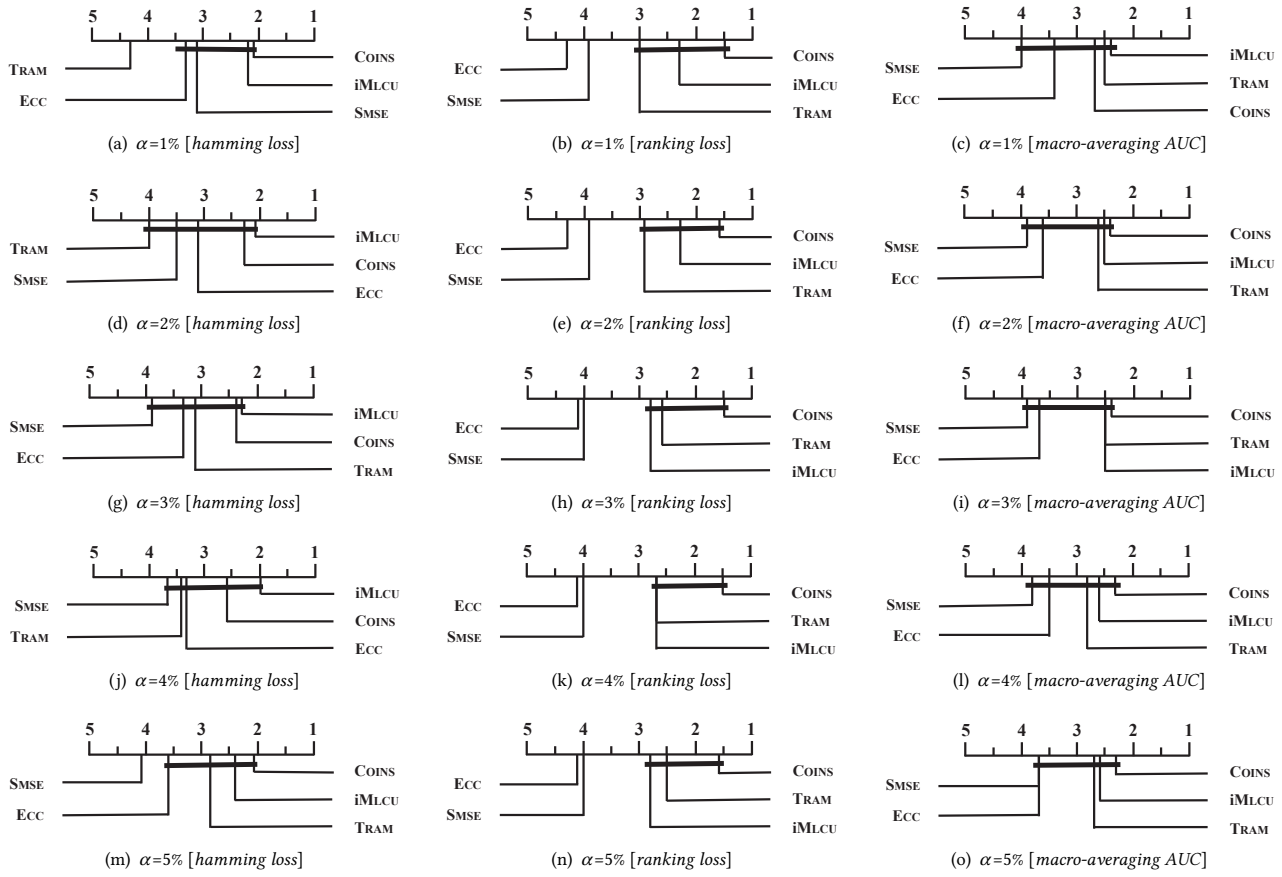

Figure 2: Comparison of Coins (control algorithm) against other comparing algorithms with the *Bonferroni-Dunn test* (transductive setting). Algorithms not connected with Coins in the CD diagram are considered to have significantly different performance from the control algorithm (CD=1.7664 at 0.05 significance level). Left column: *hamming loss*; Middle column: *ranking loss*; Right column: *macro-averaging AUC*.

[24] W. Wang and Z.-H. Zhou. 2013. Co-training with insufficient views. In *Proceedings of the 5th Asian Conference on Machine Learning*. Canberra, Australia, 467–482.

[25] F. Wu, Z. Wang, Z. Zhang, Y. Yang, J. Luo, W. Zhu, and Y. Zhuang. 2015. Weakly semi-supervised deep learning for multi-label image annotation. *IEEE Transactions on Big Data* 1, 3 (2015), 109–122.

[26] L. Wu and M.-L. Zhang. 2013. Multi-label classification with unlabeled data: An inductive approach. In *Proceedings of 5th Asian Conference on Machine Learning*. Canberra, Australia, 197–212.

[27] T. Yu and W. Zhang. 2016. Semisupervised multi-label learning with joint dimensionality reduction. *IEEE Signal Processing Letters* 23, 6 (2016), 795–799.

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

[29] M.-L. Zhang and Z.-H. Zhou. 2014. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering* 26, 8 (2014), 1819–1837.

[30] Z.-H. Zhou and M. Li. 2010. Semi-supervised learning by disagreement. *Knowledge and Information Systems* 24, 3 (2010), 415–439.

[31] X. Zhu and A. B. Goldberg. 2009. Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 3, 1 (2009), 1–130.