Multi-Label Collective Classification

Xiangnan Kong* Xiaoxiao Shi† Philip S. Yu‡

Abstract

Collective classification in relational data has become an important and active research topic in the last decade, where class labels for a group of linked instances are correlated and need to be predicted simultaneously. Collective classification has a wide variety of real world applications, *e.g.* hyperlinked document classification, social networks analysis and collaboration networks analysis. Current research on collective classification focuses on single-label settings, which assumes each instance can only be assigned with exactly one label among a finite set of candidate classes. However, in many real-world relational data, each instance can be assigned with a set of multiple labels simultaneously. In this paper, we study the problem of multi-label collective classification and propose a novel solution, called ICML (Iterative Classification of Multiple Labels), to effectively assign *a set* of multiple labels to each instance in the relational dataset. The proposed ICML model is able to capture the dependencies among the label sets for a group of related instances and the dependencies among the multiple labels within each label set simultaneously. Empirical studies on real-world tasks demonstrate that the proposed multi-label collective classification approach can effectively boost classification performances in multi-label relational datasets.

1 Introduction

Traditional machine learning and data mining approaches assume that instances are independent and identically distributed, and each testing instance is predicted with a class label independently. However, in many relational datasets [25] or information networks [8], the instances are implicitly or explicitly related, with complex dependencies. For example, in collaboration networks, the researchers who collaborate with each other are more likely to share similar research topics than researchers without any collaboration. An effective model for relational datasets should be able to consider the dependencies among the related instances during classification steps. Motivated by these challenges, *collective classification* has received considerable attention in the last decade, where a group of related instances are classified simultaneously rather than independently.

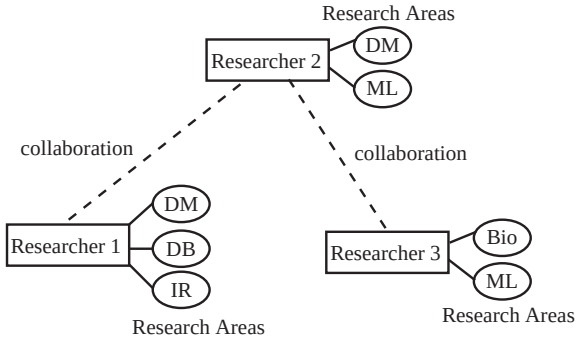


Figure 1: An example of multi-label relational data.

In the literature, collective classification problem has been extensively studied [19, 25, 15, 27]. Conventional approaches focus on single-label classification problems, which assume that each instance in the relational dataset has only one label among a finite set of candidate classes (As shown in Figure 2(a), $\boldsymbol{x}_i$ denotes the i-th instance and Y_i is its label, instances directly linked by an edge are related). However, in many real-world applications, each instances can be assigned with more than one label. For example, in collaboration networks (*e.g.* Figure 1) one author can have multiple research areas of interest, such as *machine learning*, *data mining* and *bioinformatics*; In web page networks, one hyper-linked web page may have multiple tags indicating multiple topics in the web page. The analysis and management of multi-label relational data can be significantly improved if each relational instance can be tagged with a set of multiple labels automatically. This setting is also known as multi-label classification where each instance can be associated with multiple categories (As shown in Figure 2(b), $\boldsymbol{x}_i$ denotes the i-th instance, $\{Y_i^j\}$ is the set of labels assigned to $\boldsymbol{x}_i$). It has been shown useful in many real-world applications such as text categorization [16, 23] and bioinformatics [6].

*Department of Computer Science, University of Illinois at Chicago, USA. xkong4@uic.edu.

†Department of Computer Science, University of Illinois at Chicago, USA. xiaoxiao@cs.uic.edu.

‡Department of Computer Science, University of Illinois at Chicago, USA. psyu@cs.uic.edu.

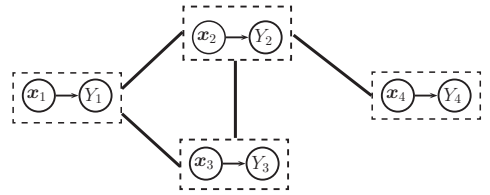
Formally, a multi-label collective classification problem corresponds to predicting the label sets of a group of *related* instances simultaneously in the label set space, *i.e.* the power set of all labels (As shown in Figure 2(c), $\boldsymbol{x}_i$ denotes the i-th instance, $\{Y_i^j\}$ is the set of labels assigned to $\boldsymbol{x}_i$ and instances directly linked by an edge are *related*). Collective classification is particularly challenging in multi-label settings. The reason is that, in the single-label settings, conventional collective classification methods can classify a group of related instances simultaneously by considering the dependencies among related instances for one label concept. But in multi-label settings, each instance can have multiple label concepts within its label set, and the dependencies among related instances with multiple labels are more complex. A group of label sets for related instances are required to be predicted simultaneously.

Despite its value and significance, the multi-label collective classification problem has not been studied in this context so far. If we consider collective classification and multi-label learning as a whole, the major research challenges on multi-label collective classification can be summarized as follows:

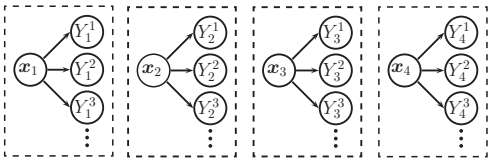
Relational Data: One fundamental problem in multi-label collective classification lies in the complex dependencies among the related instances. Conventional multi-label classification approaches assume, explicitly or implicitly, that instances are unrelated and the label sets of the testing instances are predicted independently [6, 28, 5]. However in the context of relational data, the label sets of related instances are not independent, which should be predicted simultaneously.

Multiple Labels: Another fundamental problem in multi-label collective classification lies in the multiple label concepts for each instance. Conventional collective classification approaches focus on single-labeled settings [19, 25, 15]. The learning strategy strictly follows the assumption that each instance has only one label. However, in many real-world applications, one instance can usually be assigned with multiple labels, which are not independent from each other. The complex dependencies among different labels should also be exploited in order to effectively predict a group of label sets simultaneously for the related instances.

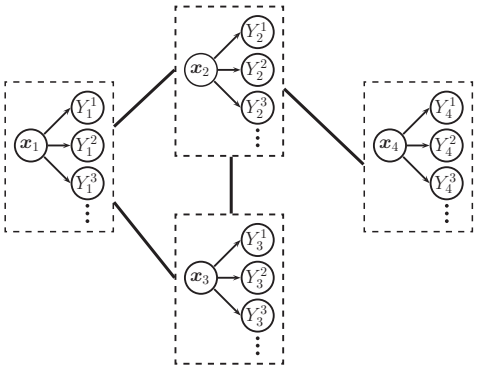
In this paper, we study the problem of multi-label collective classification and propose a novel solution, called ICML (Iterative Classification of Multiple Labels), to effectively assign *a set* of multiple labels to each instance in the relational dataset. Different from conventional collective classification methods, the proposed ICML model can exploit three types of dependencies within a multi-label relational dataset simultaneously: (1) Intra-instance cross-label dependencies; (2) Inter-instance single-label dependencies; (3) Inter-instance cross-label dependencies. By explicitly exploiting these dependencies, our ICML method can effectively assign a group of label sets for related instances simultaneously with an iterative inference procedure. Empirical studies on real-world tasks demonstrate that the proposed multi-label collective classification approach can significantly boost the multi-label classification performances in multi-label relational datasets.



(a) Collective Classification



(b) Multi-Label Classification



(c) Multi-Label Collective Classification

Figure 2: Comparison of the settings for different classification problems. Each rectangle represents an instance. $\boldsymbol{x}_i$ denotes the attributes of the instance, and Y_i or Y_i^k denotes a class label. Instances directly linked by an edge are related.

The rest of the paper is organized as follows. We start by a brief review on related works of collective classification and multi-label learning. Then we introduce the preliminary concepts, give the problem analysis in Section 3 and present the ICML algorithm in Section 4. Then Section 5 reports the experiment results. In Section 6, we conclude the paper.

Table 1: Important Notations.

Symbol	Definition
$\mathcal{V} = \{v_1, \cdots, v_n\}$	the set of nodes
$\mathcal{E} = \{e_i\}$	the set of edges or links, $e_i \in \mathcal{V} \times \mathcal{V}$
$\mathcal{N}(i)$ or $\mathcal{N}(v_i, \mathcal{E})$	the index set of all nodes directly-linked to v_i with edges in $\mathcal{E}$
$\mathcal{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$	the given attribute values for each node
$\mathcal{Y} = \{\boldsymbol{Y}_1, \cdots, \boldsymbol{Y}_n\}$	the set of variables for label sets of the nodes
$\boldsymbol{Y}_i = \left(Y_i^1, \cdots, Y_i^q\right)$	the vector of variables for the label set of node v_i, and $Y_i^k \in \{0, 1\}$
L and U	the index sets for training nodes and testing nodes, and $L \cup U = \{1, \cdots, n\}$
$\mathcal{E}_L = \{e_i\}$	the set of edges or links between training nodes, $e_i \in \mathcal{V}_L \times \mathcal{V}_L$
$\boldsymbol{y}_i = \left(y_i^1, \cdots, y_i^q\right)$	the given label set for node v_i ($i \in L$), and $Y_i^k = y_i^k$
$\boldsymbol{Y}^k = \left(Y_1^k, \cdots, Y_n^k\right)$	the vector of variables for the k-th label
$\boldsymbol{Y}_i^{\{-k\}} = \left(\cdots, Y_i^{k-1}, Y_i^{k+1}, \cdots\right)$	the vector of all the Y_i^p variables with indices $\{p \vert p \neq k\}$
$\boldsymbol{y}_i^{\{-k\}} = \left(\cdots, y_i^{k-1}, y_i^{k+1}, \cdots\right)$	the vector of all the y_i^p values with indices $\{p \vert p \neq k\}$
$\boldsymbol{Y}_{j \in \mathcal{N}(i)}^{\{-k\}}$	the vector of all the Y_j^p variables with indices $\{(p, j) \vert p \neq k, j \in \mathcal{N}(i)\}$

2 Related Work

To the best of our knowledge, this paper is the first work addressing the multi-label collective classification problem. Our work is related to both multi-label classification techniques and collective classification on relational data. We briefly discuss both of them.

Multi-label learning deals with the classification problem where each instance can belong to multiple different classes simultaneously [26, 10, 12, 7, 3, 13]. The goal of multi-label classification is to predict each instance with a set of multiple labels in the space of all label sets, *i.e.* the power set of all labels, which is exponential to the number of possible labels. In order to effectively tackle this challenging task, conventional multi-label learning approaches focus on exploiting the correlations between different labels to improve the label set prediction performances. Based upon the type of correlations among different labels exploited, conventional multi-label learning methods can be roughly categorized into three categories: (1) *binary* approaches: The first type of approaches assumes all different labels are independent, which converts the multi-label problem into multiple independent binary classification problems (one for each label) [1]. ML-KNN[29] is one of the binary methods, which extends the kNN algorithm to a multi-label version using *maximum a posteriori* (MAP) principle to determine the label set predictions. (2) *pairwise* approaches: The second type of approaches exploit the pairwise relation between different labels [9]. For example, Elisseeff and Weston [6] presented a kernel method RANK-SVM by minimizing a loss function named *ranking loss* to properly rank label pairs. (3) *High-order* approaches: The third type of approaches considers the high-order correlations among different labels. Such approaches includes random subset ensemble approaches [21, 22], Bayesian network based approach [28] and full-order approaches [4, 5, 2].

Collective classification of single-label relational data has also been investigated by many researchers. The task is to predict the classes for a group of related instances simultaneously, rather than predicting a class for each instance independently. In relational datasets, the class label of one instance can be related to the class labels (sometimes attributes) of the other related instances. Conventional collective classification approaches focus on exploiting the correlations among the class labels of related instances to improve the single-label classification performances. Roughly speaking, existing collective classification approaches can be categorized into two types based upon the different approximate inference strategies: (1) Local methods: The first type of approaches employ a local classifier to iteratively classify each unlabeled instance using both attributes of the instances and relational features derived from the related instances. This type of approaches involves an iterative process to update the labels and the relational features of the related instances, *e.g.* iterative convergence based approaches [19, 15] and Gibbs sampling approaches [18]. Many local classifiers have been used for local methods, *e.g.* logistic regression [15], Naive Bayes [19], relational dependency network [20], *etc.* (2) Global methods: The second type of approaches optimizes global objective functions on the entire relational dataset, which also uses both attributes and relational features for inference [25]. For a detailed

review of collective classification please refer to [24].

3 Problem Definition

Before presenting the collective classification model for multi-label relational data, we first introduce the notations that will be used throughout this paper.

Suppose we are given a multi-label relational dataset represented as a graph $G(\mathcal{V}, \mathcal{E}, \mathcal{X}, \mathcal{Y}, \mathcal{C})$, where $\mathcal{V} = \{v_1, \cdots, v_n\}$ is a set of nodes, $\mathcal{E}$ is a set of edges (directed or undirected) in $\mathcal{V} \times \mathcal{V}$. On each node $v_i \in \mathcal{V}$ we have a vector of attributes $\boldsymbol{x}_i \in \mathbb{R}^d$ in the d-dimensional input space, and $\mathcal{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$. Let $\mathcal{C} = \{l_1, l_2, \cdots, l_q\}$ be the finite set of q possible label concepts. On each node $v_i \in \mathcal{V}$, we also have a vector of label variables $\boldsymbol{Y}_i = \left(Y_i^1, \cdots, Y_i^q\right) \in \{0,1\}^q$ indicating the multiple labels assigned to node v_i, $\mathcal{Y} = \{Y_i\}_{i=1}^n$. Assume further that we are given a set of known values $\mathcal{Y}_L$ for nodes in a training set $\mathcal{V}_L \subset \mathcal{V}$, where L denotes the index set for training data. $\mathcal{Y}_L = \{\boldsymbol{y}_i | v_i \in \mathcal{V}_L\}$, where $\boldsymbol{y}_i = \left(y_i^1, \cdots, y_i^q\right) \in \{0,1\}^q$ is a binary vector indicating the observed multiple labels assigned to node v_i, *i.e.* $y_i^k = 1$ iff the label l_k is in the labels set of v_i, and $Y_i^k = y_i^k$ $(\forall k \in \{1, \cdots, q\})$. Then the task of multi-label collective classification is to infer the values of $\boldsymbol{Y}_i \in \mathcal{Y}_U$ for the remaining nodes in the testing set $(\mathcal{V}_U = \mathcal{V} - \mathcal{V}_L)$.

As reviewed in Section 2, in multi-label classification tasks, the inference problem is to estimate $P(\mathcal{Y}|\mathcal{X})$ given a multi-label training set. Conventional multi-label classification approaches usually require i.i.d. assumptions, the inference for each instance is performed independently:

$$P(\mathcal{Y}|\mathcal{X}) \propto \prod_{i \in U} P(\boldsymbol{Y}_i|\boldsymbol{x}_i)$$

In multi-label classification problems, the simplest solution is to decompose the multi-label classification problem into multiple binary classification problems (one for each label) by assuming different labels are independent:

$$P(\boldsymbol{Y}_i|\boldsymbol{x}_i) = \prod_{k=1}^{q} P(Y_i^k|\boldsymbol{x}_i)$$

However, in multi-label relational datasets, there are complex dependencies not only among different instances but also among different labels, which cannot be ignored. In order to solve the multi-label collective classification problem more effectively, in this paper, we explicitly consider three types of dependencies in multi-label relational datasets.

Intra-Instance Cross-Label Dependency

The first type of dependencies we consider is the correlations of different labels within each instance's label set, *i.e.* in multi-label relational datasets, different labels are actually not independent within each label set. For example, in research collaboration networks, the probability of a researcher in the area of *data mining* should be high if we know he/she is already in the area of *database* or *machine learning*; the researcher is unlikely to be in the area of *bioinformatics*, if we know he/she is already in the area of *operating system*.

Conventional multi-label learning approaches are focused on exploiting this type of dependencies, *i.e.* intra-instance cross-label dependencies, to improve the classification performances, which model $P(Y_i^k|\boldsymbol{x}_i, \boldsymbol{Y}_i^{\{-k\}})$ (Shown in Figure 3(a)). Here $\boldsymbol{Y}_i^{\{-k\}}$ denotes the vector of all the variables Y_i^p with indices $\{p : p \neq k\}$. Hence, we have

$$P(\boldsymbol{Y}_i|\boldsymbol{x}_i) = \prod_{k=1}^{q} P(Y_i^k|\boldsymbol{x}_i, \boldsymbol{Y}_i^{\{-k\}})$$

Inter-Instance Single-Label Dependency

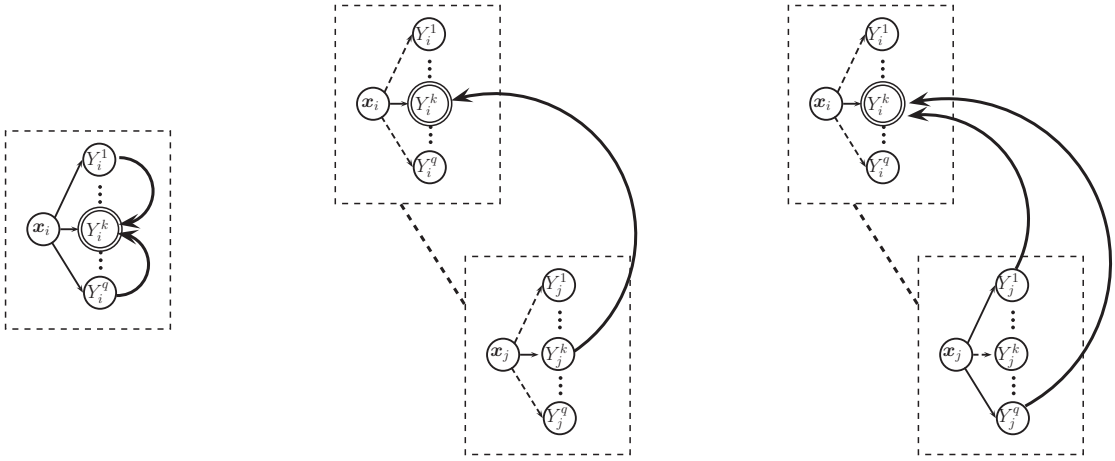
The second type of dependencies we consider is the dependencies among related instances for each single-label, *i.e.* in multi-label relational datasets, predictions on related instances are actually not independent for each label. For example, in research collaboration networks, the probability of an author in the area of *data mining* should be high if we know he/she has already collaborated with some researchers in the area of *data mining*.

Conventional collective classification approaches are focused on exploiting this type of dependencies, *i.e.* inter-instance single-label dependencies, to improve the classification performances, which models $P(Y_i^k|\boldsymbol{x}_i, \boldsymbol{Y}_{j \in \mathcal{N}(i)}^k)$ (Shown in Figure 3(b)). Here $\boldsymbol{Y}_{j \in \mathcal{N}(i)}^k$ denotes the vector containing all variable Y_j^k $(\forall j \in \mathcal{N}(i))$, and $\mathcal{N}(i)$ denotes the index set of *related* instances to the i-th instance, *i.e.*, the instances directly linked to the i-th instance. Hence, by considering inter-instance single-label dependencies alone, we will have

$$P(\boldsymbol{Y}^k|\boldsymbol{X}) = \prod_{i \in U} P(Y_i^k|\boldsymbol{x}_i, \boldsymbol{Y}_{j \in \mathcal{N}(i)}^k)$$

Inter-Instance Cross-Label Dependency

The third type of dependencies we consider is the dependencies of different labels among related instances, *i.e.* in multi-label relational datasets, different labels on related instances are not independent. For example, in research collaboration networks, the probability of an author in the area of *data mining* should be high if we know he/she has already collaborated with some researchers in the area of *database* or *machine learning*;



(a) Intra-Instance Cross-Label Dependency (b) Inter-Instance Single-Label Dependency (c) Inter-Instance Cross-Label Dependency

Figure 3: Three types dependencies in multi-label collective classification problem. Y_i^k with double circles denotes the current label variable to be predicted. Each rectangle represents an instance. $\boldsymbol{x}_i$ denotes the attribute values of the instance.

the author is unlikely to be in the area of *bioinformatics*, if we know he/she has already collaborated with some researchers in the area of *operating system*.

To the best of our knowledge, this type of dependencies, *i.e.* inter-instance cross-label dependency, has neither been studied in multi-label learning before nor in collective classification research. It models $P(Y_i^k|\boldsymbol{x}_i, \boldsymbol{Y}_{j\in\mathcal{N}(i)}^{\{-k\}})$ as shown in Figure 3(c). Hence, by considering inter-instance cross-label dependencies alone, we will have

$$P(\mathcal{Y}|\mathcal{X}) \propto \prod_{i\in U}\prod_{k=1}^{q} P(Y_i^k|\boldsymbol{x}_i, \boldsymbol{Y}_{j\in\mathcal{N}(i)}^{\{-k\}})$$

For multi-label collective classification problem, we want to perform inference on the label sets of a group of related instances simultaneously. Thus when all the three types of dependencies in the multi-label relational dataset are considered together, we have

$$P(\mathcal{Y}|\mathcal{X}) \propto \prod_{i\in U}\prod_{k=1}^{q} P(Y_i^k|\boldsymbol{x}_i, \boldsymbol{Y}_i^{\{-k\}}, \boldsymbol{Y}_{j\in\mathcal{N}(i)}^{k}, \boldsymbol{Y}_{j\in\mathcal{N}(i)}^{\{-k\}})$$

$$= \prod_{i\in U}\prod_{k=1}^{q} P(Y_i^k|\boldsymbol{x}_i, \boldsymbol{Y}_i^{\{-k\}}, \boldsymbol{Y}_{j\in\mathcal{N}(i)})$$

4 The ICML Algorithm

In classifying multi-label relational datasets, the most naïve approach of approximating $P(\mathcal{Y}|\mathcal{X}) \propto \prod_{i\in U}\prod_{k=1}^{q} P(Y_i^k|\boldsymbol{x}_i)$ with the assumptions that the instances and different labels are independent. However, this approach cannot achieve satisfactory performances, because the complex dependencies among related instances and different labels are totally ignored. In this section, we propose a simple and effective algorithm for multi-label collective classification problem. We aim to develop a model in which $P(Y_i^k|\boldsymbol{x}_i, \boldsymbol{Y}_i^{\{-k\}}, \boldsymbol{Y}_{j\in\mathcal{N}(i)})$ can be estimated. However, in general the values of $\boldsymbol{Y}_i(i \in U)$ on the testing data are not known during the inference. In this section, we propose to approximate $\boldsymbol{Y}_i(i \in U)$ with a simple iterative inference process.

Conventional collective classification based on local models, *e.g.* ICA (Iterative Classification Algorithm) [15], provide a simple but very effective method for single-label collective classification problem. Inspired by the success of the iterative inference methods, in this paper, we propose a similar framework for multi-label collective classification problem. This approach is called ICML (Iterative Classification of Multiple Labels), summarized in Figure 4.

The general idea is as follow: we model the joint probability based on the following assumption: if instance/node v_i and v_j are not directly connected, the variables Y_i^k $(k = 1, \cdots, q)$ are conditional independent from $\boldsymbol{Y}_j$ given the label sets of all v_i's neighbors. Hence the local conditional probability on label k can be modeled by a base learner with some extended *relational features* built upon the predicted $\boldsymbol{Y}_j$'s $(j \in \mathcal{N}(i))$ and the predicted $\boldsymbol{Y}_i^{\{-k\}}$. And the joint probability can be modeled based on these local conditional probabilities by treating the instances as *i.i.d.* and different labels as independent.

Input:

$\mathcal{V}$: the set of nodes, $\mathcal{E}$: the set of edges/links.
$\mathcal{X}$: attribute vectors, $\mathcal{Y}_L$: label sets for the training data.
L: the index set for training data, U: the index set for testing data.
A: a base learner for local model, Max_It: maximum # of iteration.

Training:

- Learn the local models, for $k = 1 \cdots q$:
  1. Construct an extended training set $\mathcal{D}^k = \left\{(\boldsymbol{x}_i^k, \boldsymbol{y}_i)\right\}$ by converting each instance $\boldsymbol{x}_i$ to $\boldsymbol{x}_i^k$ as follow:
     $\boldsymbol{x}_i^k = \left(\boldsymbol{x}_i, \boldsymbol{y}_i^{\{-k\}}, \text{SLRelFeature}(v_i, \mathcal{E}_L, \mathcal{Y}_L, k), \text{CLRelFeature}\left(v_i, \mathcal{E}_L, \mathcal{Y}_L, k\right)\right)$
  2. Let $f^k = A(\mathcal{D}^k)$ be the local model for the k-th label.

Bootstrap:

- Estimate the label sets, for $i \in U$
  1. Produce an estimated values $\hat{\boldsymbol{Y}}_i = \left(\hat{Y}_i^1, \cdots, \hat{Y}_i^q\right)$ for $\boldsymbol{Y}_i$ as follow:
     $\hat{Y}_i^k = f^k\left((\boldsymbol{x}_i, \boldsymbol{0})\right)$ using attributes only, for $k = 1, \cdots, q$.

Iterative Inference:

- Repeat until convergence or #iteration$> Max_It$
  1. Construct the extended testing instance by converting each instance $\boldsymbol{x}_i$ to $\boldsymbol{x}_i^k$ $(i \in U)$ as follow:
     $\boldsymbol{x}_i^k = \left(\boldsymbol{x}_i, \hat{\boldsymbol{Y}}_i^{\{-k\}}, \text{SLRelFeature}(v_i, \mathcal{E}, \mathcal{Y}_L \cup \{\hat{Y}_i | i \in U\}, k), \text{CLRelFeature}(v_i, \mathcal{E}, \mathcal{Y}_L \cup \{\hat{\boldsymbol{Y}}_i | i \in U\}, k)\right)$
  2. Update the estimated values $\hat{\boldsymbol{Y}}_i$ for $\boldsymbol{Y}_i$ on each testing instance $(i \in U)$ as follow:
     $\hat{Y}_i^k = f^k(\boldsymbol{x}_i^k)$, for $k = 1, \cdots, q$.

Output:

$\hat{\boldsymbol{Y}}_i = \left(\hat{Y}_i^1, \cdots, \hat{Y}_i^q\right)$: The label set for each test instance $(i \in U)$.

Figure 4: The ICML algorithm

In collective classification, one practical difficulty in modeling the dependencies of related instances is that each instance may be linked with different number of instances as neighbors. In order to build a fixed number of relational features for each instance, we employs *aggregation* functions to combine the predictions on the label sets of related instances. In this paper, we use the fraction of the related instances which have the label l_k in their label sets as the relational feature for label l_k. Other aggregation functions, such as COUNT and MODE aggregators [15], can also be used. In detail, given an aggregation function, we can get two different types of relational features from the label sets of related instances as shown in Figure 5, *i.e.* "SLRelFeature" for inter-instance single-label dependencies and "CLRelFeature" for inter-instance cross-label dependencies.

In the spirit of ICA framework [15, 17, 18], the inference procedure of our ICML method has two parts: *bootstrap* and *iterative inference* as shown in Figure 4. (1) At the beginning of the inference procedure, the label sets of all the unlabeled instances are unknown.

- $\boldsymbol{x}_{\text{SL}} = \text{SLRelFeature}\left(v, \mathcal{E}, \{\boldsymbol{Y}_i\}, k\right)$
  1. Get related instances for node v in edge set $\mathcal{E}$, *i.e.* the related index set $\mathcal{C} = \mathcal{N}(v, \mathcal{E})$
  2. $\boldsymbol{x}_{\text{SL}} = Aggregation\left(\{Y_i^k | i \in \mathcal{C}\}\right)$

- $\boldsymbol{x}_{\text{CL}} = \text{CLRelFeature}\left(v, \mathcal{E}, \{\boldsymbol{Y}_i\}, k\right)$
  1. Get related instances for node v in edge set $\mathcal{E}$, *i.e.* the related index set $\mathcal{C} = \mathcal{N}(v_i, \mathcal{E})$
  2. $\boldsymbol{x}_{\text{CL}} = Aggregation\left(\{\boldsymbol{Y}_i^{\{-k\}} | i \in \mathcal{C}\}\right)$

Figure 5: The functions for constructing inter-instance single-label relational features (SLRelFeature) and inter-instance cross-label relational features (CLRelFeature)

The *bootstrap* part is used to assign an initial label set for each instance using only attributes of each node. In our current implementation, we simply initialize the relational features for unlabeled instances with all zero

vectors. Other strategies for *bootstrap* part can also be used in this framework, *e.g.* training another set of local models on training data using attributes only, and then we use these local models to predict the initial label sets for unlabeled data. (2) In the *iterative inference* part, we iteratively update the relational features based on the predictions of local models and update the prediction of local models using the newly updated relational features on each instance. The iterative process stops when the predictions of all local models are stabilized or a maximum number of iteration has been reached. In our current implementation, we update the $\boldsymbol{Y}_i$'s for $(r+1)$-th iteration (say $\boldsymbol{Y}_i^{(r+1)}$) using the predicted values in the r-th iteration ($\boldsymbol{Y}_j^{(r)}$) only. Other strategies for *iterative inference*, such as sequentially updating $\boldsymbol{Y}_i^{(r+1)}$ based on the latest predictions for $\boldsymbol{Y}_j (j \in \mathcal{N}(i))$ on the related instances with a certain order, can also be adopted into our framework.

5 Experiments

5.1 Data Collections In order to evaluate the multi-label collective classification performances, we tested our algorithm on three real-world multi-label collective classification datasets (Summarized in Table 2).

DBLP-A Dataset: The first dataset studied in this paper is extracted from the DBLP database[1]. DBLP provides bibliographic information on computer science journals and proceedings. We extracted a DBLP co-authorship network containing authors who have published as least 2 papers during the years 2000-2010 as the nodes of the network. We linked any two authors who have collaborated with each other. On each node, we extracted a bag-of-words representation of all the paper titles published by the author to use as attributes of the node. The words with frequencies less than 1 % are remove from the vocabulary. Each author can have multiple research topics of interests from the 6 research areas given below. We select some most representative conferences from each of the area. The selected conference list for each research area is as follows:

- **Database:** ICDE, VLDB, SIGMOD, PODS, EDBT
- **Data Mining:** KDD, ICDM, SDM, PKDD, PAKDD
- **Artificial Intelligence:** IJCAI, AAAI
- **Information Retrieval:** SIGIR, ECIR
- **Computer Vision:** CVPR
- **Machine Learning:** ICML, ECML

[1] http://www.informatik.uni-trier.de/~ley/db/

Table 2: Summary of experimental datasets. "Ave. Card" denotes the average cardinality of label sets assigned to the nodes.

	Data Sets		
Characteristics	DBLP-A	DBLP-B	IMDB
# Features	687	699	586
# Nodes	4638	4559	4081
# Links	16447	14407	41087
# Class	6	6	22
Ave. Card	1.70	1.24	2.13

If the author has published papers in any of these conferences, we assume the author is interested in the corresponding research areas. The task is to classify each author with a set of multiple research areas of interest.

DBLP-B Dataset: The second dataset studied is also from DBLP database. We extract 6 different research areas and the selected conferences as follows:

- **Algorithms & Theory:** STOC, FOCS, SODA, COLT
- **Natural Language Processing:** ACL, ANLP, COLING
- **Bioinformatics:** ISMB, RECOMB
- **Networking:** SIGCOMM, MOBICOM, INFOCOM
- **Operating Systems:** SOSP, OSDI
- **Distributed & Parallel Computing:** PODC, ICS

The same setups with DBLP-A are also used here to build the collaboration network.

IMDB Dataset: The third dataset studied in this paper is from the IMDB database[2], which contains information about directors, plots of a movie, *etc.* The classification task is to predict the movie's genre (the movie type, *e.g.*, horror, comedy). Each movie can be assigned with a subset of multiple movie genres among 27 candidate genres in the IMDB dataset. We extracted 4081 movies released in the US between 2000 and 2010 and used a bag-of-word representation for the plots of a movie as the attributes. The words with frequencies less than 2 % are remove from the vocabulary. Movies directed by the same director are linked together. After removing the movies with less than 10 links, we have a dataset with 4081 movies and 22 candidate genres. Detailed properties of the dataset can be found in Table 2.

[2] http://www.imdb.com/interfaces

Table 3: Summary of compared methods.

Method	Type of Classification	Dependencies Exploited	Publication
BSVM	Binary Classification	All Independent	[1]
CC	Multi-Label Classification	① Intra-Instance Cross-Label Dependency	[22]
ECC	Multi-Label Classification	① Intra-Instance Cross-Label Dependency	[22]
SL-ICA	Collective Classification	② Inter-Instance Single-Label Dependency	[15]
ML-ICA	Multi-Label Collective Classification	② Inter-Instance Single-Label Dependency ③ Inter-Instance Cross-Label Dependency	This paper
ICML	Multi-Label Collective Classification	① Intra-Instance Cross-Label Dependency ② Inter-Instance Single-Label Dependency ③ Inter-Instance Cross-Label Dependency	This paper

5.2 Evaluation Metrics The performance evaluation for multi-label classification requires more complicated criteria than conventional single-label classification problems. Here we adopt some metrics used in [9, 11, 14, 30, 5] to evaluate the classification performance in a multi-label relational data. Assume we have a multi-label relational dataset $\mathcal{D}_U$ containing n multi-label instances $(\boldsymbol{x}_i, \boldsymbol{Y}_i)$, where $\boldsymbol{Y}_i \in \{0,1\}^q$ $(i = 1, \cdots, n)$. Let $h(\boldsymbol{x}_i)$ denote a multi-label classifier's predicted label set for $\boldsymbol{x}_i$. We have the following evaluation criteria:

a) Hamming loss [5, 30]: Hamming loss is one of the most frequently used criterion, which evaluates the number of labels whose relevance is incorrectly predicted.

$$HammingLoss(h, \mathcal{D}_U) = \frac{1}{n}\sum\nolimits_{i=1}^{n}\frac{1}{q}\|h(\boldsymbol{x}_i) \oplus \boldsymbol{Y}_i\|_1$$

where $\oplus$ stands for the symmetric difference of two sets (XOR operation), and $\|\cdot\|_1$ denotes the l_1-norm. The smaller the value, the better the performance.

b) Subset 0/1 Loss [5, 9]: evaluates strictly whether a classifier's label set prediction is *exactly* correct.

$$\text{SubsetLoss}(h, \mathcal{D}_U) = \frac{1}{n}\sum_{i=1}^{n} I\left(h(\boldsymbol{x}_i) \neq \boldsymbol{Y}_i\right)$$

where $I(\cdot)$ denotes the indicator function, *i.e.* $I(\pi) = 1$ iff π holds, otherwise $I(\pi) = 0$. The smaller the value, the better the performance.

c) Micro F1 [9, 11, 14]: evaluates a classifier's label set prediction performance, which considers both micro average of Precision and Recall on different classes with equal importance.

$$\text{micro-F1}(h, \mathcal{D}_U) = \frac{2 \times \sum_{i=1}^{n} \|h(\boldsymbol{x}_i) \cap \boldsymbol{Y}_i\|_1}{\sum_{i=1}^{n} \|h(\boldsymbol{x}_i)\|_1 + \sum_{i=1}^{n} \|\boldsymbol{Y}_i\|_1}$$

The larger the value, the better the performance.

d) Macro F1 [9]: evaluates a classifier's label set prediction performance, which considers average the F1 measure on the predictions of different labels.

$$\text{macro-F1}(h, \mathcal{D}_U) = \frac{1}{q}\sum_{k=1}^{q}\frac{2 \times \sum_{i=1}^{n} h^k(\boldsymbol{x}_i) Y_i^k}{\sum_{i=1}^{n} h^k(\boldsymbol{x}_i) + \sum_{i=1}^{n} Y_i^k}$$

where Y_i^k is the k-th entry of $\boldsymbol{Y}_i$ and $h^k(\boldsymbol{x}_i)$ is the k-th entry of $h(\boldsymbol{x}_i)$. The larger the value, the better the performance.

All experiments are conducted on machines with Intel Xeon$^{\text{TM}}$Quad-Core CPUs of 2.26 GHz and 24 GB RAM.

5.3 Compared Methods In order to demonstrate the effectiveness of our multi-label collective classification approach, we test with following methods (summarized in Table 3):

- Multi-Label Collective Classification (ICML): We first test our proposed method, ICML (Iterative Classification of Multiple Labels), for multi-label collective classification. The proposed approach can exploit all the three types of dependencies for multi-label collective classification.
- Multi-label relational feature + ICA (ML-ICA): This method is extended from the ICA (Iterative

Table 4: Results (mean±std (rank)) on the DBLP-A dataset. "↓" indicates the smaller the value the better the performance; "↑" indicates the larger the value the better the performance.

Methods	Evaluation Criteria								Ave. Rank
	Hamming Loss↓		Subset 0/1 Loss↓		micro-F1 ↑		macro-F1 ↑		
ICML	0.162±0.006	(1)	0.600±0.017	(1)	0.657±0.012	(1)	0.634±0.012	(1)	1
ML-ICA	0.174±0.007	(2)	0.659±0.012	(2)	0.633±0.015	(2)	0.614±0.016	(2)	2
SL-ICA	0.332±0.007	(4)	0.916±0.006	(3)	0.323±0.013	(4)	0.246±0.010	(5)	4
ECC	0.379±0.005	(6)	0.924±0.009	(4.5)	0.351±0.009	(3)	0.291±0.012	(3)	4.125
CC	0.374±0.006	(5)	0.924±0.005	(4.5)	0.310±0.006	(5)	0.262±0.003	(4)	4.625
BSVM	0.327±0.011	(3)	0.931±0.008	(6)	0.258±0.010	(6)	0.190±0.008	(6)	5.25

Table 5: Results (mean±std (rank)) on the DBLP-B dataset. "↓" indicates the smaller the value the better the performance; "↑" indicates the larger the value the better the performance.

Methods	Evaluation Criteria								Ave. Rank
	Hamming Loss↓		Subset 0/1 Loss↓		micro-F1 ↑		macro-F1 ↑		
ICML	0.060±0.006	(1)	0.293±0.023	(1)	0.835±0.016	(1)	0.804±0.018	(1)	1
ML-ICA	0.071±0.003	(2)	0.354±0.009	(2)	0.802±0.012	(2)	0.769±0.013	(2)	2
SL-ICA	0.212±0.004	(3)	0.917±0.015	(5)	0.169±0.021	(5)	0.084±0.009	(5)	4.5
ECC	0.257±0.017	(6)	0.782±0.056	(4)	0.382±0.046	(3)	0.169±0.017	(3)	4
CC	0.248±0.018	(5)	0.742±0.049	(3)	0.342±0.048	(4)	0.143±0.014	(4)	4
BSVM	0.215±0.004	(4)	0.949±0.006	(6)	0.117±0.015	(6)	0.067±0.014	(6)	5.5

Table 6: Results (mean±std (rank)) on the IMDB dataset. "↓" indicates the smaller the value the better the performance; "↑" indicates the larger the value the better the performance.

Methods	Evaluation Criteria								Ave. Rank
	Hamming Loss↓		Subset 0/1 Loss↓		micro-F1 ↑		macro-F1 ↑		
ICML	0.104±0.001	(2)	0.846±0.004	(1.5)	0.487±0.003	(1)	0.417±0.013	(1.5)	1.5
ML-ICA	0.094±0.003	(1)	0.846±0.012	(1.5)	0.483±0.013	(2)	0.417±0.015	(1.5)	1.5
SL-ICA	0.109±0.002	(3.5)	0.892±0.006	(4)	0.420±0.009	(4)	0.334±0.013	(4)	3.875
ECC	0.109±0.006	(3.5)	0.883±0.015	(3)	0.455±0.010	(3)	0.360±0.012	(3)	3.125
CC	0.120±0.004	(6)	0.899±0.012	(6)	0.388±0.016	(6)	0.297±0.016	(6)	6
BSVM	0.111±0.002	(5)	0.894±0.014	(5)	0.417±0.006	(5)	0.325±0.011	(5)	5

Classification Algorithm) [15] by adding relational features according to *inter-instance-cross-label* dependencies for multi-label collective classification. The only difference from ICML is that ML-ICA does not consider the *intra-instance-cross-label* dependencies.

- Binary decomposition + ICA (SL-ICA): We compare with another baseline using a binary decomposition method similar to [1]: The multi-label relational dataset is first divided into multiple single-label relational datasets by one-vs-all binary decomposition. For each binary classification task, we use the conventional collective classification method, ICA [15] to classify the relational data.

- Multi-Label Classification (CC):We also compare with another baseline: multi-label classification method using classifier chain [22]. *Intra-instance-cross-label* dependencies are explicitly consider in this method.

Table 7: Performance of ICML with different iterations on the DBLP-A dataset. "↓" indicates the smaller the value the better the performance; "↑" indicates the larger the value the better the performance.

	Evaluation Criteria			
Iteration	Hamming Loss↓	Subset 0/1 Loss↓	micro-F1 ↑	macro-F1 ↑
$r = 0$	0.1807±0.0061	0.6563±0.0184	0.5736±0.0190	0.5520±0.0260
$r = 1$	0.1654±0.0068	0.6130±0.0172	0.6408±0.0190	0.6175±0.0246
$r = 2$	0.1623±0.0052	0.6050±0.0178	0.6547±0.0142	0.6309±0.0203
$r = 3$	0.1616±0.0047	0.6037±0.0187	0.6581±0.0124	0.6344±0.0193
$r = 4$	0.1615±0.0046	0.6033±0.0197	0.6591±0.0117	0.6354±0.0185
$r = 5$	0.1616±0.0047	0.6035±0.0198	0.6591±0.0118	0.6352±0.0186
$r = 6$	0.1615±0.0047	0.6035±0.0198	0.6593±0.0118	0.6355±0.0186
$r = 7$	0.1615±0.0047	0.6033±0.0199	0.6594±0.0117	0.6355±0.0186
$r = 8$	0.1615±0.0047	0.6033±0.0199	0.6594±0.0117	0.6355±0.0186

Table 8: Results of ICML with different iterations on the DBLP-B dataset. "↓" indicates the smaller the value the better the performance; "↑" indicates the larger the value the better the performance.

	Evaluation Criteria			
Iteration	Hamming Loss↓	Subset 0/1 Loss↓	micro-F1 ↑	macro-F1 ↑
$r = 0$	0.0782±0.0040	0.3694±0.0217	0.7715±0.0150	0.7206±0.0154
$r = 1$	0.0621±0.0022	0.2999±0.0154	0.8289±0.0077	0.7826±0.0135
$r = 2$	0.0605±0.0019	0.2933±0.0151	0.8347±0.0067	0.7901±0.0131
$r = 3$	0.0603±0.0019	0.2931±0.0153	0.8353±0.0067	0.7910±0.0123
$r = 4$	0.0603±0.0019	0.2931±0.0153	0.8353±0.0067	0.7910±0.0123

- Multi-Label Classification + Ensemble (ECC): This baseline method is an ensemble of classifier chains (CC) [22]. The ensemble is created by training different classifier chains using randomly sampled subset of instances with random label orders.

- Binary SVM (BSVM): This baseline[1] assumes all the label prediction are independent. Binary decomposition is used, on each binary classification task (one for each label) SVM is used as the base classifier. Then the predictions of SVMs for all labels are combined to make to final prediction.

For fair comparison, LibSVM [2] with linear kernel is used as the base classifier for all the compared methods. The maximum number of iteration in the methods ICML, ML-ICA and SL-ICA are all set as 10.

5.4 Performances of Multi-Label Collective Classification We first study the effectiveness of the proposed ICML method on multi-label collective classification. In our experiment, 5-fold cross validation is performed on each experimental data set to evaluate the multi-label collective classification performances. We report the average and standard deviation results. Table 4, Table 5 and Table 6 show the performances of each of the six models on three datasets according to the four evaluation criteria. Performance ranks of each model on each of the evaluation criteria are also listed.

In all these datasets, we can observe that the baseline BSVM, which treats all predictions as independent, is outperformed by all the other methods which explicitly consider dependencies among predications from various aspects. In general, these results can support the importance of exploiting the different types of dependencies in multi-label relational datasets. For example, SL-ICA can improve performances over BSVM by exploiting the inter-instance single-label dependencies. ECC and CC can also outperform BSVM by exploiting the intra-instance cross-label dependencies. Similar results have also been reported both in collective classification literatures and multi-label learning researches.

Then we observe that in all these datasets our multi-label collective classification methods (both ICML and ML-ICA) can outperform the multi-label classification baselines which only exploit the intra-instance cross-label dependencies and assume instances are inde-

pendent. These results support our first assumption that in multi-label relational datasets, the label sets of related instances are not independent. Exploiting the complex dependencies among the related instances (*i.e.* inter-instance single-label dependencies and inter-instance cross-label dependencies) can boost the performance of the multi-label classification in relational datasets.

We further observe that our multi-label collective classification methods (both ICML and ML-ICA) outperform the binary decomposition method based on single-label collective classification (SL-ICA), which exploit the dependencies among related instances while assuming different labels are independent. These results support our second assumption that in multi-label relational datasets, the multiple labels are not independent with complex dependencies among them. Exploiting the dependencies among different labels (*i.e.* intra-instance cross-label dependencies and inter-instance cross-label dependencies) can further improve the multi-label classification performance in relational datasets.

Another observation we get from these results is that in most of these datasets the proposed ICML method outperforms the baseline ML-ICA, whose implementation is similar to ICML while ignoring the intra-instance cross-label dependencies. This result can further support the importance of exploiting intra-instance cross-label dependencies in our ICML method because the multiple labels within each label set are correlated . The performance of ML-ICA is comparable with ICML on the IMDB dataset, because ML-ICA can implicitly and indirectly exploit intra-instance cross-label dependencies by considering both inter-instance cross-label dependencies and inter-instance single-label dependencies, however, on all the other two datasets ICML can further improve the performance over ML-ICA by explicitly exploiting the intra-instance cross-label dependencies.

5.5 Sensitivity of Iteration Number In our model, an iterative procedure is used for label set inferences. In Table 7 and Table 8, we show the convergence rate of the proposed multi-label collective classification approach (ICML) by testing the performances after each iteration step. We observe that the iterative inference of ICML converges after seven iterations in DBLP-A dataset and four iterations in DBLP-B dataset. The performance of the ICML converge very fast after the first few iterations. From these results, we can see that the performance of ICML is not sensitive to the parameter Max_It (maximum number of iterations) as long as Max_It is assigned with a modest number. In our default setting, we use 10 as the default maximum number of iterations.

6 Conclusion

In this paper, we studied the problem of multi-label collective classification. Different from conventional collective classification approaches in relational datasets which assume each instance can only be assigned with exactly one label among a finite set of candidate classes, in multi-label collective classification problems, each instance can have a subset of multiple labels among the candidate label set. We propose a novel solution to multi-label collective classification, called ICML (Iterative Classification of Multiple Labels), to effectively assign a set of multiple labels to each instance in the relational dataset. The proposed ICML model is able to exploit the dependencies among the label sets for a group of related instances and the dependencies among the multiple labels within each label set simultaneously. Empirical studies on real-world tasks demonstrate that the proposed multi-label collective classification approach can effectively boost classification performances in multi-label relational datasets.

7 Acknowledgements

This work is supported in part by NSF through grants IIS 0905215, DBI-0960443, OISE-0968341 and OIA-0963278.

References

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

[2] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[3] G. Chen, Y. Song, F. Wang, and C. Zhang. Semi-supervised multi-label learning by solving a sylvester equation. In *Proceedings of the 8th SIAM International Conference on Data Mining*, pages 410–419, Atlanta, GA, 2008.

[4] W. Cheng and E. Hüllermeier. Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2-3):211–225, 2009.

[5] K. Dembczyński, W. Cheng, and E. Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of the 28th International Conference on Machine Learning*, pages 279–286, Haifa, Israel, 2010.

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

[7] I. Vlahavas G. Tsoumakas. Random k-labelsets: An ensemble method for multilabel classification. In *Pro-*

ceedings of the 18th European Conference on Machine Learning, pages 406–417, Warsaw, Poland, 2007.

[8] J. Gao, F. Liang, W. Fan, C. Wang, Y. Sun, and J. Han. Community outliers and their efficient detection in information networks. In *Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 913–822, Washington, DC, 2010.

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

[10] S. Godbole and S. Sarawagi. Discriminative methods for multi-labeled classification. In *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 22–30, Sydney, Australia, 2004.

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

[12] H. Kazawa, T. Izumitani, H. Taira, and E. Maeda. Maximal margin labeling for multi-topic text categorization. In *Advances in Neural Information Processing Systems 15*, pages 649–656. 2005.

[13] X. Kong and P. S. Yu. Multi-label feature selection for graph classification. In *Proceedings of the 10th IEEE International Conference on Data Mining*, pages 274–283, Sydney Australia, 2010.

[14] Y. Liu, R. Jin, and L. Yang. Semi-supervised multi-label learning by constrained non-negative matrix factorization. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 421–426, Boston, MA, 2006.

[15] Q. Lu and L. Getoor. Link-based classification. In *Proceedings of the 20th International Conference on Machine Learning*, pages 496–503, Washington, DC, 2003.

[16] A. McCallum. Multi-label text classification with a mixture model trained by EM. In *AAAI'99 Workshop on Text Learning*, Orlando, FL, 1999.

[17] L. K. McDowell, K. M. Gupta, and D. W. Aha. Cautious inference in collective classification. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 596–601, Vancouver, Canada, 2007.

[18] L. K. McDowell, K. M. Gupta, and D. W. Aha. Cautious collective classification. *Journal of Machine Learning Research*, 10:2777–2836, 2009.

[19] J. Neville and D. Jensen. Iterative classification in relational data. In *AAAI'10 Workshop on Learning Statistical Models from Relational Data*, Austin, TX, 2000.

[20] J. Neville and D. Jensen. Collective classification with relational dependency networks. In *KDD'03 Workshop on Multi-Relational Data Mining*, pages 77–91, Washington, DC, 2003.

[21] J. Read, B. Pfahringer, and G. Holmes. Multi-label classification using ensembles of pruned sets. In *Proceedings of the 8th IEEE International Conference on Data Mining*, pages 995–1000, Pisa, Italy, 2008.

[22] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. In *Proceedings of the 20th European Conference on Machine Learning*, pages 254–269, Bled, Slovenia, 2009.

[23] R. E. Schapire and Y. Singer. Boostexter: a boosting-based system for text categorization. *Machine Learning*, 39(2-3):135–168, 2000.

[24] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[25] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *UAI*, pages 482–492, Edmonton, Alberta, 2002.

[26] N. Ueda and K. Saito. Parametric mixture models for multi-labeled text. In *Advances in Neural Information Processing Systems 13*, pages 721–728. 2003.

[27] P. S. Yu, J. Han, and C. Faloutsos. *Link Mining: Models, Algorithms and Applications*. Springer, 2010.

[28] M.-L. Zhang and K. Zhang. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 999–1008, Washington, DC, 2010.

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

[30] X. Zhang, Q. Yuan, S. Zhao, W. Fan, W. Zheng, and Z. Wang. Multi-label classification without the multi-label cost. In *Proceedings of the 10th SIAM International Conference on Data Mining*, pages 778–789, Columbus, OH, 2010.