Capturing Joint Label Distribution for Multi-Label Classification through Adversarial Learning

Shangfei Wang, *Senior Member, IEEE,* Guozhu Peng, and Zhuangqiang Zheng

Abstract—Label correlations are important for multi-label learning. Although current multi-label learning approaches can exploit first-order, second-order, and high-order label dependencies, they fail to exploit complete label correlations, which are included in the joint label distribution of the ground truth labels. However, directly modeling the complex and unknown joint label distribution is very challenging, if not impossible. In this paper, we propose an adversarial learning framework to enforce similarity between joint distribution of the ground truth multi-labels and the predicted multiple labels. Specifically, the proposed multi-label learning method includes a multi-label classifier and a label discriminator. The classifier minimizes error between predicted labels and corresponding ground truth labels and gives the discriminator room for error. The object of the discriminator is to distinguish the predicted labels from the ground truth labels. The classifier and discriminator are trained simultaneously through an alternate process. By adversarial learning, the joint label distribution of the predicted multi-labels converges to the joint distribution inherent in the ground truth multi-labels, and thus boosts the performance of multi-label learning as demonstrated in the experiments on eleven benchmark databases.

Index Terms—multi-label learning, joint label distribution, adversarial learning

1 Introduction

Traditional supervised learning paradigms assume that each instance is associated with one class label. However, in real-world applications, a single instance is commonly associated with multiple labels. For example, a scenic image may contain trees, flowers, and mountains; a variety show video may convey joy as well as tenderness; a movie includes categories like costume, plot, and setting. Multi-label learning has attracted increasing attention in recent years due to its widespread application in areas such as **image annotation ([1], [2], [3], [4])**, text categorisation ([5], [6]), the emotion detection of music ([7], [8]), and social network mining ([9], [10]).

Successfully exploiting label correlations is crucial to deal with the overwhelming size of the output space [11], which is the biggest challenge of multi-label learning. Crucial information exists in label correlations. For example, if the label “sun” is present, the probability of the appearance of “blue sky” would be high. A film is unlikely to be labelled “terrifying” if it’s already labelled ”comedy”. The former instance would be a coexistent relationship; the latter is an example of mutually exclusive relations. Complex relations among labels, which are difficult to manually specify, are inherent in the ground truth labels.

Current multi-label learning work directly captures label correlations from ground truth labels. The exploited label dependencies include first-order, second-order, and high-order label correlations. However, they are far from enough to capture label correlations effectively. Compared with exploiting correlation between two or more labels, modelling joint distribution of multiple labels, which takes all kinds of label correlations into account, is a more thoroughly way. For example, the coexistent and mutually exclusive relations mentioned above are the conditional distribution of one label under another label, which can be regard as one part of the joint distribution of all labels. The distribution must be measured in a certain way to enforce statistical similarity between the predicted labels and the ground truth labels. However, directly capturing the complex and unknown joint label distribution for ground truth labels is quite challenging.

Only recently, a few works have used a probabilistic graph model to model label distribution for multi-label learning. For example, [12], [13], and [14] leverage a bayesian network (BN) to model joint label distribution by decomposing the joint label distribution into the product of conditional distributions; [15], [16], and [17] leverage a restricted boltzmann machine (RBM) model to capture joint label distribution through a hidden layer. Unfortunately, both BN- and RBM-based methods make assumptions. A BN assumes that some nodes are conditionally independent in order to limit network complexity. An RBM model assumes an explicit form of joint label distribution. These assumptions may not be suitable for some applications, thus limiting the performance of model.

In this paper, we propose to exploit label correlations by enforcing distribution similarity between the predicted labels and the ground truth labels. Unlike a BN or RBM, which captures joint label distribution from ground truth labels first and then fits the predicted labels of a basic classifier into the captured joint label distribution, we make the distribution of the predicted labels similar to the distribution of ground truth labels directly through an adversarial framework inspired by generative adversarial nets (GANs) [18]. Specifically, in addition to learning a multi-

- *S. Wang, G. Peng, and Z. Zheng are with the Department of of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui, 230027, China. E-mail: sfwang@ustc.edu.cn; gzpeng@mail.ustc.edu.cn; zheng001@mail.ustc.edu.cn*

Manuscript received April 19, 2005; revised August 26, 2015.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2019.2922603, IEEE Transactions on Knowledge and Data Engineering

label classifier C, we introduce a label discriminator D. The object of discriminator D is to distinguish whether the input label is the ground truth label or the predicted label from C. The multi-label classifier C has two objects: the first is to minimize the error between the predicted labels and the corresponding ground truth labels, and the second is to "fool" D, i.e., output labels that were wrongly discriminated as ground truth labels by D. Through this adversarial framework, we minimize the traditional supervised loss and achieve distribution similarity between the predicted labels and the ground truth labels.

The rest of the paper is organized as follows. In Section 2, we introduce the related works on multi-label learning with a focus on exploiting label correlations. In Section 3, we state the problem and give some notations. In Section 4, we propose our multi-label learning method to exploit label correlations by leveraging joint label distribution. In Section 5, sufficient experiments are conducted on several widely used databases. The last section concludes our paper.

2 Related Work

2.1 Multi-Label Learning

Multi-label learning methods can be divided into three types according to the degree of label correlations exploited: first-order, second-order, and high-order. These methods also fall into one of two categories [11]: problem transformation methods, which tackle the multi-label learning problem by transforming it into another well-established learning scenario; and algorithm adaptation methods, which adapt popular learning techniques to deal with multi-label data directly. Next, we briefly summarize the multi-label classification methods according to the degree of label correlations exploited.

First-order methods handle the classification of each label independently, thus ignoring the correlations among labels. The most well-known first-order method is *binary relevance* (BR) [19], which decomposes the original multi-label classification problem into several independent binary classification problems (one per label). Other first-order methods include *multi-label k-nearest neighbor* (ML-kNN) [20], which adapts *k-nearest neighbor* to deal with multi-label data; *multi-label decision tree* (ML-DT) [21], which adapts a decision tree (like C4.5) to deal with multi-label data; and *predictive clustering trees* (PCTs) [22], which form a general framework for prediction that can be instantiated to a particular prediction task by defining a distance metric and prototype. The performance of first-order methods may be inferior as it ignores label correlations.

Second-order methods exploit pairwise label relations, i.e., when one label is related to another one. Common second-order methods include *ranking by pairwise comparison* (RPC) [23], which transforms the task of multi-label learning into the task of label ranking and transforms the multi-label dataset into $\frac{L(L-1)}{2}$ (where L is the number of labels) binary label datasets; *calibrated label ranking* (CLR) [24], which extends RPC by introducing a calibrated label and using a majority voting strategy at prediction time; *quick weighted voting algorithm* (QWeighted) [25], which introduces a more effective voting strategy than the majority voting used by the CLR method; *QWeighted for multi-label classification* (QWML) [26], which adapts QWeighted by repeating the process until all relevant labels are determined; *ranking support vector machine* (Rank-SVM) [27] which adapts kernel methods to deal with multi-label data, *multi-label back-propagation*, (BP-MLL) [28] which adapts a popular back-propagation algorithm to deal with multi-label data; and *collective multi-label classifier* (CML) [29], which adapts a maximum entropy principle to deal with multi-label data. Second-order methods exploit label correlations to some extent, thus achieving a better generalization performance than first-order methods in most cases.

High-order methods exploit high-order correlations among labels and assume a label is influenced by more than one other label. Common high-order methods include *random k-labelsets* (RAkEL) [30], which transforms the task of multi-label learning into the task of multi-class classification; *hierarchy of multi-label classifiers* (HOMER) [31], which transforms multi-label classification task with a large set of labels into a tree-shaped hierarchy of simpler multi-label classification tasks; and *classifier chains* (CC) [32], which transforms the task of multi-label learning into a chain of binary classification tasks. The correlation-modeling ability of high-order methods is stronger than first- and second-order methods.

Many multi-label learning methods can obtain better generalization performance through an ensemble method. RAkEL, for example, is actually an ensemble method handling several multi-class classification problems, each of which use a random subset of labels. Using CC as basic classifier, *ensembles of classifier chains* (ECC) [32] trains a set of CC classifiers with a random chain ordering and a random subset of training samples. Through random forest, decision tree-based multi-label learning methods can be extended to ensemble methods *Random forest of predictive clustering trees* (RF-PCT) [33] is an ensemble that uses PCTs, and *random forest of ML-C4.5* (RFML-C4.5) [34] is an ensemble that uses the ML-C4.5 tree. An extensive review of multi-label classification methods can be found in [34], [35], [36], [37], [38], and [11].

Although high-order methods can exploit high-order relations among labels, they fail to consider all label correlations. In CC, for example, the binary classifier of each label is only trained with the correlations between the current label and all previous labels. The label correlations exploited by CC are limited by the chain ordering. While ECC can train several CC classifiers with different chain orders, it is unfeasible to consider every possible chain order. Instead of exploiting certain kinds of label correlations, some probabilistic graph-based methods model joint label distribution for multi-label learning. Some works use a bayesian network to decompose the joint label distribution into the product of conditional distributions ([12], [13], and [14]). In a BN structure, nodes can be related to each other indirectly or may be conditionally independent. In real-life scenarios, it is difficult to say that two labels are completely independent, so BN cannot preserve all correlations among labels. Additionally, BN is a directed graph model and assumes that the labels are related in the form of a hierarchy, such as "steamship" and "sea"; "grassland" and "Africa". This hierarchical structure may not be suitable for some applications. For example, a scenic image may contain a car, house, and mountain, which are not subject to a hierarchy.

To address the problems above, some works use an undirected graph model, i.e., an RBM model, to model the joint distribution of multiple labels through a hidden layer ([15], [16], and [17]). However, RBM-based models assume an explicit form of joint distribution (i.e., the explicit form of energy function), which is not suitable for all applications. Additionally, RBM-based methods use a complex inference procedure.

In this paper, we exploit label correlation by using joint label distribution, which contains all label correlations. We train a multi-label classifier by enforcing distribution similarity between the predicted labels and the ground truth labels. It's important to note that the label distribution we consider is different from the label distribution used in Label Distribution Learning (LDL) [39]. In our work, one instance has several binary labels, while in [39], one instance is assigned several real values, representing the degree to which each label describes the instance. Therefore, the label distribution we consider is the joint distribution of all labels, since the population contains all training instances. The label distribution in [39] is only for one instance.

2.2 Generative Adversarial Nets

Goodfellow *et al.* [18] introduced the generative adversarial nets (GANs), which consists of a generator $\mathcal{G}$ and a discriminator $\mathcal{D}$. This framework actually implements a two-player zero sum game between $\mathcal{G}$ and $\mathcal{D}$. Specifically, $\mathcal{D}$ tries to distinguish whether the input sample is from a real data distribution or generated by generator, and $\mathcal{G}$ tries to "fool" $\mathcal{D}$, i.e. to make $\mathcal{D}$ wrongly classify a generated sample as real data. The object function of this two-player zero sum game is shown as Equation 1

$$\begin{aligned}\min_{\mathcal{G}}\max_{\mathcal{D}} V(\mathcal{D},\mathcal{G}) =&\mathbb{E}_{x\sim P_{data}(x)}\big[\log \mathcal{D}(x)\big]\\ &+\mathbb{E}_{z\sim P_z(z)}\big[\log(1-\mathcal{D}(\mathcal{G}(z))\big]\end{aligned}\tag{1}$$

where $P_{data}(x)$ is the real data distribution and $P_z(z)$ is an arbitrary noisy distribution. It had been proven in [18] that Eq. 1 gets the global optimum only when the generator's distribution $P_g = P_{data}$. The discriminator $\mathcal{D}$ and generator $\mathcal{G}$ can be trained through an alternate optimization process, first fixing $\mathcal{G}$ and optimizing $\mathcal{D}$, then fixing $\mathcal{D}$ and optimizing $\mathcal{G}$, and repeating this process until convergence. The details and variants of GAN can be found in [40], [41], and [42].

In this paper, we propose an adversarial multi-label learning framework similar to GAN in order to achieve distribution similarity between the predicted labels and the ground truth labels. Compared to related works, our contributions are as follows:

- We are the first to propose an adversarial framework for multi-label learning.
- Through adversarial learning, we implicitly capture the joint distribution of multiple labels without any assumptions and achieve significant improvements on eleven benchmark databases.

3 Problem Statement

Let $\boldsymbol{U} = \{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^N$ be the training set for multi-label learning, in which a feature vector $\boldsymbol{x}_n \in \mathbb{R}^q$ is associated with multiple class ground truth labels $\boldsymbol{y}_n = \{y_n^1, y_n^2, ..., y_n^L\}$. q is the dimension of feature vector $\boldsymbol{x}$, L is the number of labels, and N is the number of instances. Each $y_n^j (1 \leq j \leq L) \in \{0,1\}$. Let $\boldsymbol{X} = \{\boldsymbol{x}_n\}_{n=1}^N$ denote the feature set including all features in $\boldsymbol{U}$ and $\boldsymbol{Y} = \{\boldsymbol{y}_n\}_{n=1}^N$ denote the ground truth label set including all ground truth labels in $\boldsymbol{U}$. Given the training set $\boldsymbol{U}$, our goal is to learn a multi-label classifier $\mathrm{C}\colon \mathbb{R}^q \rightarrow \{0,1\}^L$. If we do not consider the label correlations, minimizing the traditional cross-entropy loss $\mathcal{L}_{ce}$ is a direct way to learn the multi-label classifier $\mathrm{C}(\boldsymbol{x}) = round(F(\boldsymbol{x}))$, where $F(\boldsymbol{x})$ is the output of a sigmoid activation.

Label correlations are curial for multi-label learning. In this paper, we explore label correlations by enforcing distribution similarity between the predicted labels and the ground truth labels, i.e., minimizing the distance $d(P_p, P_g)$ between the joint distribution of the predicted labels P_p and the joint distribution of the ground truth labels P_g. Motivated by this goal, and considering the basic supervised loss, we purpose the full objective of this paper as follows:

$$\min_{\Phi}(1-\gamma) * \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\boldsymbol{U}}\mathcal{L}_{ce}(F(\boldsymbol{x};\Phi),\boldsymbol{y}) + \gamma * d(P_p, P_g),\tag{2}$$

where Φ is the parameters vector of the multi-label classifier and γ is the trade-off rate between the first and the second term. $\mathcal{L}_{ce}$ is the supervised loss shown as follows:

$$\begin{aligned}\mathcal{L}_{ce}(F(\boldsymbol{x};\Phi),\boldsymbol{y}) = -\Big[&\boldsymbol{y}^\top \log F(\boldsymbol{x};\Phi) +\\ &(\boldsymbol{1}-\boldsymbol{y})^\top \log(\boldsymbol{1}-F(\boldsymbol{x};\Phi))\Big]\end{aligned}\tag{3}$$

We do not directly model distribution P_p and P_g, since the modeling processes are complex and errors could occur. We use an adversarial framework to achieve the goal of minimizing $d(P_p, P_g)$.

4 Proposed Approach

Inspired by GAN, we propose an adversarial framework that enforces similar distributions of the predicted labels and the ground truth labels while minimizing the errors between predicted labels and their corresponding ground truth labels. In the framework of original GAN, the generator generates "realistic" data from the random noise space. In this paper, the generator is replaced by the multi-label classifier C which recognizes the multiple labels of instances from the feature space. The discriminator D gives the probability that each input label vector comes from the ground truth label set $\boldsymbol{Y}$. The target for the discriminator D is to give a high probability for ground truth labels in $\boldsymbol{Y}$ and a low probability for the predicted labels from C. The target for the multi-label classifier C is the opposite. The competition game between the multi-label classifier C and the discriminator D can be represented by the following optimization problem. (For differentiability of objectives, we input $F(\boldsymbol{x})$ to D directly):

$$\min_{\Phi}\max_{\Psi}\mathbb{E}_{\boldsymbol{y}'\sim\boldsymbol{Y}}\log\mathrm{D}(\boldsymbol{y}';\Psi)+\mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{X}}\log(1-\mathrm{D}(F(\boldsymbol{x};\Phi);\Psi))\tag{4}$$

Where $\boldsymbol{y}'$ represents the ground truth label sampled from $\boldsymbol{Y}$, and Φ and Ψ are parameter vectors of the multi-label classifier and label discriminator, respectively. From Equation 4, we can achieve the objective of minimizing $d(P_p, P_g)$ in Equation 2. Combining the objectives in Equations 3 and 4 with the trade-off rate γ, i.e., replacing $d(P_p, P_g)$ in Equation 2 with the objective in Equation 4, we obtain the whole objective of the proposed approach as:

$$\begin{aligned}\min_{\Phi}\max_{\Psi}\ &\gamma\big[\mathbb{E}_{\boldsymbol{y}'\sim\boldsymbol{Y}}\log \mathrm{D}(\boldsymbol{y}';\Psi)\\ &+\mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{X}}\log(1-\mathrm{D}(F(\boldsymbol{x};\Phi);\Psi))\big]\\ &+(1-\gamma)\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\boldsymbol{U}}\mathcal{L}_{ce}(F(\boldsymbol{x};\Phi),\boldsymbol{y})\end{aligned} \tag{5}$$

From the above equation, we find that if we set $\gamma = 0$, it degrades into a traditional supervised problem that doesn't consider label correlations. If we set $\gamma = 1$, it only leverages the label distribution information. We should seek the optimum balance.

We can rewrite Equation 5 as Equation 6 to let $F(\boldsymbol{x};\Phi)$ in one square bracket, which helps to find individual objective for F.

$$\begin{aligned}\min_{\Phi}\max_{\Psi}\ &\gamma\mathbb{E}_{\boldsymbol{y}'\sim\boldsymbol{Y}}\log \mathrm{D}(\boldsymbol{y}';\Psi)\\ &+\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\boldsymbol{U}}\big[\gamma\log(1-\mathrm{D}(F(\boldsymbol{x};\Phi);\Psi))\\ &+(1-\gamma)\mathcal{L}_{ce}(F(\boldsymbol{x};\Phi),\boldsymbol{y})\big]\end{aligned} \tag{6}$$

The objectives in Equations 5 and 6 can not be optimized directly. Following the optimization procedure of original GAN, we update the multi-label classifier and label discriminator alternately shown in Fig. 1. Specifically, while updating label discriminator D, the multi-label classifier C is fixed and we randomly sample a mini-batch of feature vectors from $\boldsymbol{X}$ and a mini-batch of ground truth labels from $\boldsymbol{Y}$. Both the predicted labels and the ground truth labels are inputted to discriminator D, and D tries to correctly distinguish between them. While updating multi-label classifier C, the label discriminator D is fixed and we randomly sample a mini-batch of training instances including feature vectors and corresponding ground truth labels from training set $\boldsymbol{U}$. The predicted labels are inputted to discriminator D. C tries to minimize the cross-entropy loss between the predicted labels and the corresponding ground truth labels, and lets D make mistakes simultaneously.

We need to find the individual objectives for the multi-label classifier and label discriminator. From the updating procedure of label discriminator D and Equation 5, we extract the term containing D in Equation 5 as the objective for discriminator D shown as Equation 7, where parameters Φ are fixed.

$$\begin{aligned}\min_{\Psi} V_{\mathrm{D}}(\mathrm{C},\mathrm{D}) = -\big[&\mathbb{E}_{\boldsymbol{y}'\sim\boldsymbol{Y}}\log \mathrm{D}(\boldsymbol{y}';\Psi)\\ &+\mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{X}}\log(1-\mathrm{D}(\mathrm{C}(\boldsymbol{x};\Phi);\Psi))\big]\end{aligned} \tag{7}$$

Similarly, from the updating procedure of multi-label classifier C and Equation 6, we extract the term containing F in Equation 6 as the objective for discriminator C shown as Equation 8, where parameters Ψ are fixed.

$$\begin{aligned}\min_{\Phi} V_{\mathrm{C}}(\mathrm{C},\mathrm{D}) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\boldsymbol{U}}\big[&\gamma\mathcal{L}^{\mathrm{C}}_{adv}(F(\boldsymbol{x};\Phi);\Psi)\\ &+(1-\gamma)\mathcal{L}_{ce}(F(\boldsymbol{x};\Phi),\boldsymbol{y})\big]\end{aligned} \tag{8}$$

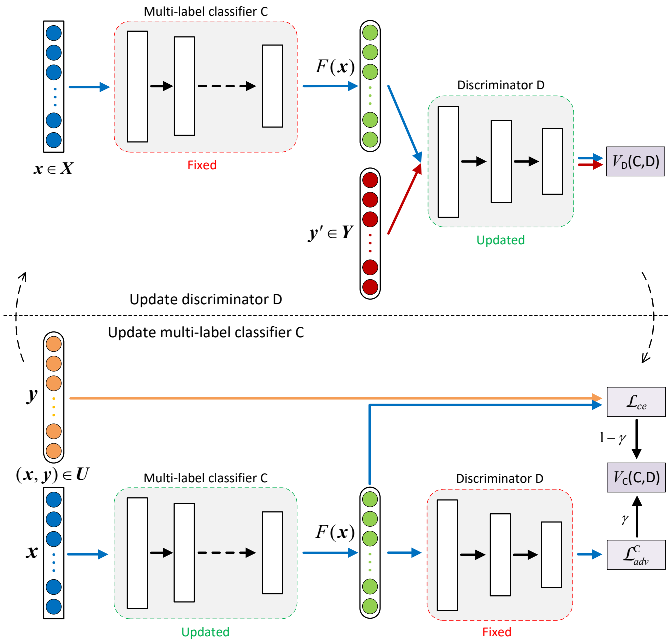


Fig. 1: The training procedure of the proposed framework. The multi-label classifier and label discriminator are updated alternately. Upper: update discriminator D. Lower: update multi-label classifier C.

Algorithm 1 Adversarial multi-label learning

```
Input: The training set $\boldsymbol{U}$, max number of training steps $T$, number of C that updates per step ($K_{\mathrm{C}}$), number of D that updates per step ($K_{\mathrm{D}}$), sampling size $S$, hyper parameter $\gamma$.
Output: The multi-label classifier C.
Initialize parameters of multi-label classifier $\Phi$ and parameters of label discriminator $\Psi$.
for $t = 1, 2, ..., T$ do
  for $k_{\mathrm{D}} = 1, ..., K_{\mathrm{D}}$ do
    Sample mini-batch of $S$ samples $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_S\}$ from feature instance set $\boldsymbol{X}$.
    Sample mini-batch of $S$ labels $\{\boldsymbol{y}'_1, \boldsymbol{y}'_2, ..., \boldsymbol{y}'_S\}$ from the ground truth label set $\boldsymbol{Y}$.
    Update the label discriminator by descending its gradient:
      $\nabla_{\Psi}\left(-\frac{1}{S}\sum_{i=1}^{S}\big[\log \mathrm{D}(\boldsymbol{y}'_i;\Psi)+\log(1-\mathrm{D}(F(\boldsymbol{x}_i;\Phi);\Psi))\big]\right)$
  end for
  for $k_{\mathrm{C}} = 1, ..., K_{\mathrm{C}}$ do
    Sample mini-batch of $S$ samples $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), ..., (\boldsymbol{x}_S, \boldsymbol{y}_S)\}$ from training set $\boldsymbol{U}$.
    Update the multi-label classifier by descending its gradient:
      $\nabla_{\Phi}\left(-\frac{1}{S}\sum_{i=1}^{S}\big[\gamma\log \mathrm{D}(F(\boldsymbol{x}_i;\Phi);\Psi)-(1-\gamma)\mathcal{L}_{ce}(F(\boldsymbol{x}_i;\Phi),\boldsymbol{y}_i)\big]\right)$
  end for
end for
```

Where $\mathcal{L}^{\mathrm{C}}_{adv}(F(\boldsymbol{x};\Phi);\Psi) = \log(1 - \mathrm{D}(F(\boldsymbol{x};\Phi);\Psi))$. In practice, it's better to minimize $-\log \mathrm{D}(F(\boldsymbol{x};\Phi);\Psi)$ rather than $\log(1 - \mathrm{D}(F(\boldsymbol{x};\Phi);\Psi))$ to avoid the problem of vanishing multi-label classifier gradients [18], so we adjust $\mathcal{L}^{\mathrm{C}}_{adv}(F(\boldsymbol{x};\Phi);\Psi) = -\log \mathrm{D}(F(\boldsymbol{x};\Phi);\Psi)$.

The updating procedures of the multi-label classifier and label discriminator in Fig. 1 are described as Algorithm 1. Multi-layer perceptron is used for the structure of both the multi-label classifier C and the label discriminator D. Any gradient-based learning rule could be used to update parameters for the optimization method. We use the Adam algorithm [43] in our experiments and implement the approach using the TensorFlow framework [44].

5 Experiments

In this section, experiments on eleven databases from different domains are conducted to demonstrate the effectiveness of the proposed approach.

5.1 Databases

Eleven databases from five different domains are used in our experiments, nine of which can be downloaded from MuLan library [45][1]. All of these databases are commonly used in multi-label classification tasks. From the image domain, we use four databases: the Corel5k database [46] , the Scene database [19], the NUS-WIDE database [47], and the VOC2007 database [48]. The Corel5k database contains Corel images that are segmented using normalized cuts. Each image can be associated with several of 374 possible labels. The Scene database contains 2407 images, and each image may include one or more of six kinds of scenery, including beach, sunset, field, fall foliage, mountain, and urban. **The NUS-WIDE dataset includes a set of images crawled from Flickr, together with their associated tags, as well as the ground-truth for 81 concepts for these images. [47]. We use 128-D cVLAD+ features described in [49]. The VOC2007 database is from the PASCAL VOC2007 challenge. The image in this database is associated with twenty object classes.**

From the text domain, we have three databases: the Enron database [50], the Medical database [51], and the Tmc2007 database [52]. The Enron database contains 1702 emails from 150 senior Enron officials. Each email can be categorized into 53 types, which can be further categorized into four groups: coarse genre, included/forwarded information, primary topics, and messages with emotional tone. The Medical database contains documents briefly summarizing the symptom history of patients. Each document is annotated with possible diseases. The Tmc2007 database contains instances of aviation safety reports, and the labels of each report are the problems described.

From the biology domain, we have two databases: the Yeast database [27] and the Eukaryote [53]. The Yeast database contains instances of genes which can be associated with one or more of 14 biological labels. **The Eukaryote database contains 7766 sequences for eukaryote species. Both the GO (Gene ontology) features and PseAAC features are provided. We use the latter.**

TABLE 1: Detailed information of eleven databases. ("**domain**" is the domain of the database. "**samples**" is the number of samples. "**features**" is the dimensionality of features. "**labels**" is the total number of labels. "**cardinality**" is the average number of labels per sample. "**density**" is equal to **cardinality/labels**. "**diversity**" is the number of distinct label sets appeared in the data set. [11])

Database	**domain**	**samples**	**features**	**labels**	**cardinality**	**density**	**diversity**
Corel5k [46]	image	5000	499	374	3.522	0.009	3175
Emotions [7]	music	593	72	6	1.869	0.311	27
Enron [50]	text	1702	1001	53	3.378	0.064	753
Mediamill [54]	video	43907	120	101	4.376	0.043	6555
Medical [51]	text	978	1449	45	1.245	0.028	94
Scene [19]	image	2407	294	6	1.074	0.179	15
Tmc2007 [52]	text	28596	49060	22	2.158	0.098	1341
Yeast [27]	biology	2417	103	14	4.237	0.303	198
NUS-WIDE [49]	image	269648	128	81	1.869	0.023	18430
Eukaryote [53]	biology	7766	440	22	1.146	0.052	112
VOC2007 [48]	image	9963	-	20	1.560	0.078	308

The Emotions database [7] contains 593 pieces of music, each of which can be associated with one or more of six emotion labels: amazement, happiness, relaxation, quietness, sadness, and anger. The Mediamill database [54] contains data about annotated videos. The label space consists of 101 semantic concepts in video, such as building, flag, horse, map, office, food, etc. Detailed information of the eleven databases is shown in Table 1.

Since these databases have been pre-divided into training and testing sets, we use the split in our experiments and extract part of the training set as the validation set. In the Tmc2007 database, the feature dimensionality of the samples is too high and the number of samples is relatively small, so we use the top 500 features as Tsoumakas et al [30] did to decrease computational complexity and make the problem learnable. In order to reduce the impact of randomness, all experiments on all databases are conducted ten times.

5.2 Metrics

To sufficiently evaluate the performances of our method, four evaluation metrics that are widely used in multi-label classification are adopted in our experiment, including two example-based evaluation metrics (accuracy and F1) and two label-based evaluation metrics (micro F1 and macro F1).

Let $S = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) | 1 \leq i \leq m\}$ be the testing set and $\boldsymbol{z}_i$ be the predicted labels corresponding to $\boldsymbol{x}_i$. The detailed definition of the four evaluation metrics are shown as follows [37]:

$$\text{Accuracy} = \frac{1}{m}\sum_{i=1}^{m}\frac{|\boldsymbol{y}_i \cap \boldsymbol{z}_i|}{|\boldsymbol{y}_i \cup \boldsymbol{z}_i|}$$

$$\text{F1} = \frac{1}{m}\sum_{i=1}^{m}\frac{2 \times |\boldsymbol{y}_i \cap \boldsymbol{z}_i|}{|\boldsymbol{y}_i| + |\boldsymbol{z}_i|}$$

$$\text{Micro F1} = \frac{2\sum_{j=1}^{L}\sum_{i=1}^{m} y_i^j z_i^j}{\sum_{j=1}^{L}\sum_{i=1}^{m} y_i^j + \sum_{j=1}^{L}\sum_{i=1}^{m} z_i^j}$$

$$\text{Macro F1} = \frac{1}{L}\sum_{j=1}^{L}\frac{2\sum_{i=1}^{m} y_i^j z_i^j}{\sum_{i=1}^{m} y_i^j + \sum_{i=1}^{m} z_i^j}$$

[1] http://mulan.sourceforge.net/datasets-mlc.html

Where z_i^j is the j-th component of $\boldsymbol{z}_i$. For four metrics above, higher values represent better performance.

5.3 Experimental Settings

On the VOC2007 database, since there is no provided extracted feature, we adopt three kinds of network as the structure of multi-label classifier, i.e., Alexnet, Vgg19, and Resnet50. For each kind of network, a pre-trained model is used. The proposed methods using three networks are named as OURS-A, OURS-V, and OURS-R, respectively. To show the performances of the compared multi-label methods on the VOC2007 database, we use the output of the pre-trained Resnet50 as the extracted features.

On other ten databases, since the extracted features are provided by database constructors, and the dimension of features are not very high, we use the multi-layer perception as the structure of the multi-label classifier. Specifically, we use the four-layers feedforward net on the Tmc2007 database and the three-layers feedforward net on the other nine databases according to the performances on the validation set. We use L1 regularization on the Corel5k, Enron, Mediamill, Medical, and Tmc2007 databases, and L2 regularization on other six databases. We use a Gaussian normalization for each dimension of features for data preprocessing.

As the selection of hyper parameters in Algorithm 1, a grid search strategy is used and the parameters which achieve the best performance on validation set are used. Specifically, for the maximum number of training steps on the Emotions and Medical databases, $T \in \{500, 600, 700, 800, 900, 1000\}$, on other databases, $T \in \{1000, 1500, 2000, 2500, 3000\}$. For the number of C that updates per step, $K_{\mathrm{C}} \in \{1, 2, 3\}$. For the number of D that updates per step, $K_{\mathrm{D}} \in \{1, 2, 3\}$. For sampling size on the Emotions and Medical databases, $S \in \{100, 200, 300\}$, on other databases, $S \in \{100, 200, 300, 400, 500\}$. For weight coefficient γ, we first select it from $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. We find the proposed method achieves best performance when $\gamma = 0.1$ on most databases. For these databases, we then select γ from $\{0.1, 0.01, 0.001, 0.0001\}$.

5.4 Comparisons

The proposed method (OURS) is compared to the following multi-label learning methods:

1) The proposed method without exploiting label correlations (OURS-W). This comparison is to analyze the contribution of the adversarial loss. We conduct the ablation study by setting the hyper parameter $\gamma = 0$.
2) Nine widely used multi-label learning methods. We compare OURS to the following first-order methods: ML-C4.5, ML-kNN, BR, and RF-PCT. We compare it to the second-order CLR method. For the high-order methods, we compare it to CC, HOMER, RAkEL, and ECC. Detailed classification of these nine methods is shown in Table 2. Madjarov *et al.* conducted experiments of the above nine algorithms on eight databases in [34], and we adopt the same experimental conditions, so we can copy their experimental results directly. **For the NUS-WIDE, Eukaryote, and VOC2007 databases, we conduct the experiments of the above nine algorithms using the provided code in MuLan library.**

TABLE 2: The multi-label learning algorithms compared with the proposed approach.

	algorithm adaptation	problem transformation	ensemble
first-order	ML-C4.5 [21], ML-kNN [20]	BR [19]	RF-PCT [33]
second-order	-	CLR [24]	-
high-order	-	CC [32], HOMER [31]	RAkEL [30], ECC [32]

3) **Zhu *et al.*'s work, multi-label learning with GLObal and loCAL label correlation (GLOCAL) [55]. GLOCAL exploits global and local label correlations simultaneously, through the global manifold regularizer and the local manifold regularizer. Since Zhu *et al.* used different databases and metrics, we conduct the experiments of GLOCAL on all databases using the provided code[2].**
4) Wang *et al.*'s work, BN [14], which also captures the joint label distribution. They tried three basic classifiers: Label Powerset (LP) [36], BR, and RAkEL. For a fair comparison, we select BR+BN since both BR and OURS-W ignore label correlations. Three metrics: accuracy, F1, and micro F1, that are adopted in both our work and theirs, are used for comparison.
5) Two RBM-based methods: three-layers RBM (TRBM) [16] and four-layers RBM (FRBM) [17]. We only compare to these two methods on the Emotion database, since it is the only common database used in ours and theirs [16], [17]. However, since BN [14], TRBM [16], and FRBM [17] adopt a ten-fold cross-validation strategy (which is different from ours), these comparisons are only for reference.
6) **Li *et al.*'s work, Conditional Graphical Lasso (CGL) [56]. CGL learns image-dependent conditional label structures base on graphical lasso framework. We only compare to CGL on the VOC2007 database.**

5.5 Results and Analyses

The results of OURS and OURS-W are listed in Table 3. **On the VOC2007 database, OURS and OURS-W use Resnet50 as the structure of classifier.** A one-sided t-test at 95% significance level was used to evaluate the superiority of OURS compared to OURS-W. From Table 3, we find that OURS performs significantly better than OURS-W on almost all databases and metrics. On the Emotions database, the Scene database, and the Yeast database for example, the experimental results of OURS on accuracy are 1.57%, 3.06%, and 3.32% higher than those of OURS-W, and on F1, the results are 1.14%, 3.94%, and 3.60% higher, respectively. For

[2] http://lamda.nju.edu.cn/code_Glocal.ashx

TABLE 3: Comparison between OURS-W and OURS. The proposed method OURS exploits label correlations, but OURS-W does not. The letter in parentheses indicates whether OUR is significantly better than OURS-W (one-sided t-test at 5% significance level.) T represents true and F represents false.

database	algorithms	accuracy	F1	micro F1	macro F1
Emotions	OURS-W	0.5650±0.0020	0.6464±0.0015	0.6944±0.0017	0.6869±0.0012
	OURS	0.5807±0.0030(T)	0.6578±0.0010(T)	0.7036±0.0026(T)	0.6966±0.0031(T)
Scene	OURS-W	0.6908±0.0010	0.7073±0.0004	0.7461±0.0003	0.7492±0.0005
	OURS	0.7214±0.0027(T)	0.7462±0.0022(T)	0.7599±0.0024(T)	0.7659±0.0019(T)
Yeast	OURS-W	0.5099±0.0002	0.6146±0.0002	0.6414±0.0001	0.3576±0.0013
	OURS	0.5432±0.0001(T)	0.6506±0.0002(T)	0.6658±0.0001(T)	0.3992±0.0001(T)
Medical	OURS-W	0.7361±0.0023	0.7669±0.0018	0.7924±0.0020	0.3456±0.0002
	OURS	0.7498±0.0005(T)	0.7776±0.0001(T)	0.8012±0.0001(T)	0.3417±0.0039(F)
Enron	OURS-W	0.4354±0.0119	0.5418±0.1268	0.5519±0.0072	0.1700±0.0003
	OURS	0.4519±0.0015(T)	0.5585±0.0013(T)	0.5670±0.0021(T)	0.1618±0.0010(F)
Corel5k	OURS-W	0.1029±0.0015	0.1457±0.0014	0.1826±0.0021	0.0241±0.0003
	OURS	0.1276±0.0008(T)	0.1805±0.0010(T)	0.2160±0.0003(T)	0.0287±0.0004(T)
Tmc2007	OURS-W	0.9860±0.0004	0.9894±0.0003	0.9914±0.0002	0.9876±0.0000
	OURS	0.9867±0.0003(F)	0.9903±0.0003(T)	0.9917±0.0002(F)	0.9890±0.0006(T)
Mediamill	OURS-W	0.4320±0.0007	0.5484±0.0002	0.5663±0.0003	0.0670±0.0007
	OURS	0.4362±0.0000(T)	0.5526±0.0001(T)	0.5724±0.0001(T)	0.0713±0.0009(T)
NUS-WIDE	OURS-W	0.3671±0.0006	0.2299±0.0017	0.3981±0.0023	0.1151±0.0019
	OURS	0.3774±0.0006(T)	0.2803±0.0028(T)	0.4424±0.0019(T)	0.1438±0.0006(T)
Eukaryote	OURS-W	0.3434±0.0040	0.3910±0.0042	0.4616±0.0030	0.1227±0.0022
	OURS	0.3645±0.0061(T)	0.4125±0.0070(T)	0.4617±0.0087(F)	0.1334±0.0043(T)
VOC2007	OURS-W	0.7160±0.0031	0.7673±0.0036	0.7787±0.0020	0.7525±0.0034
	OURS	0.7266±0.0014(T)	0.7829±0.0018(T)	0.7820±0.0012(T)	0.7616±0.0022(T)

TABLE 4: Comparisons among three deep networks on the VOC2007 database.

	Accuracy	F1	Micro F1	Macro F1
OURS-A	0.577	0.645	0.666	0.609
OURS-V	0.704	0.761	0.765	0.742
OURS-R	**0.727**	**0.783**	**0.782**	**0.762**

label-based metrics, the experimental results of OURS on micro F1 are 0.92%, 1.38%, and 2.44% higher than those of OURS-W respectively, and on macro F1, the results are 0.97%, 1.67%, and 4.16% higher than those of OURS-W, respectively.

We also achieve significant improvements on other databases in most cases. OURS-W ignores the constraint of joint label distribution that contains all label correlations. Unlike OURS-W, OURS takes advantage of the label correlations by introducing an adversarial model that can make the distribution of the predicted labels similar to the distribution of the ground truth labels. Additionally, the adversarial loss in OURS can be regarded as a regularization term, which can further confine the searching space of learning parameters to find a better locally optimal solution.

Table 3 shows that OURS does not make a significant improvement over OURS-W on the Tmc2007 database, Compared to OURS-W, OURS only achieves 0.07% improvement in accuracy, 0.09% improvement in F1, 0.03% improvement in micro F1, and 0.14% improvement in macro F1. This may be because OURS-W has already been able to achieve nearly perfect classification performances. (Almost all testing samples have been correctly classified; four evaluation metrics are close to 99%), and the distribution of the predicted labels has been very close to the distribution of the ground truth labels. The adversarial model in OURS cannot significantly improve upon this performance.

On the VOC2007 database, we use three kinds of deep network as the structure of multi-label classifier, i.e., Alexnet, Vgg19, and Resnet50. The comparisons among three networks are shown in Table 4. We can find that OURS-R performs best and OURS-A performs worst, demonstrating the superiority of Resnet50.

5.6 Comparison to Related Work

Comparisons among the proposed method and related works are shown in Tables 5, 6, 7, and 8. From these results, we can obtain following observations.

First, when comparing nine multi-label methods, it's easy to find that high-order methods outperform second-order methods and second-order methods outperform first-

TABLE 5: The performances of multi-label approaches in terms of accuracy metric. In the first column, the content in square bracket represents the degree of label correlations that the algorithm exploits. In detail, “1”, “2”, and “m” represent the first-order, the second-order, and the high-order method, respectively. Besides, the word “full” indicates the method exploits all label correlations. The number in round brackets shows the ranking of the algorithm. Bold numbers indicate the best performance. The last column “mean rank” represents the average ranking of an algorithm on eleven databases. The experiments of ML-C4.5 on the NUS-WIDE database have not been completed within one month under the available resources. These notes are appropriate for Table 6, 7, and 8.

	Emotions	Scene	Yeast	Medical	Enron	Corel5k	Tmc2007	Mediamill	NUS-WIDE	Eukaryote	VOC2007	mean rank
BR[1]	0.361(8)	0.689(6)	0.520(7)	0.206(11)	0.446(5)	0.030(5)	0.891(4)	0.403(6)	0.277(6)	0.119(9)	0.677(8)	6.818
CC[m]	0.356(10)	0.723(3)	0.527(5)	0.211(10)	0.334(10)	0.030(6)	0.899(3)	0.390(7)	0.293(3)	0.325(2)	0.687(6)	5.909
CLR[2]	0.361(9)	0.686(7)	0.524(6)	0.656(6)	0.459(3)	**0.195(1)**	0.889(5)	0.095(10)	0.278(4)	0.113(10)	0.684(7)	6.182
HOMER[m]	0.471(4)	0.717(5)	**0.559(1)**	0.713(3)	**0.478(1)**	0.179(2)	0.888(6)	0.413(5)	0.231(8)	0.184(7)	0.487(9)	4.636
ML-C4.5[1]	0.536(2)	0.569(10)	0.480(9)	0.730(2)	0.418(8)	0.002(9)	0.110(11)	0.052(11)	-	0.237(5)	0.469(10)	7.700
ML-*k*NN[1]	0.319(11)	0.629(8)	0.492(8)	0.528(9)	0.319(11)	0.014(7)	0.574(9)	0.421(4)	0.317(2)	0.064(11)	0.703(3)	7.545
RA*k*EL[m]	0.419(7)	0.734(2)	0.531(4)	0.673(5)	0.428(7)	0.000(11)	0.852(7)	0.337(9)	0.278(4)	0.127(8)	0.695(5)	6.273
ECC[m]	0.432(5)	**0.735(1)**	0.546(2)	0.611(7)	0.462(2)	0.001(10)	0.808(8)	0.349(8)	0.265(7)	0.288(3)	0.699(4)	5.182
RF-PCT[1]	0.519(3)	0.541(11)	0.478(10)	0.591(8)	0.416(9)	0.009(8)	0.914(2)	**0.441(1)**	0.129(10)	0.190(6)	0.325(11)	7.182
GLOCAL[2]	0.432(5)	0.624(9)	0.351(11)	0.690(4)	0.442(6)	0.122(4)	0.559(10)	0.437(2)	0.185(9)	0.265(4)	0.718(2)	6.000
BR+BN[m]	0.552	0.668	0.521	-	-	-	-	-	-	-		
OURS[full]	**0.581(1)**	0.721(4)	0.543(3)	**0.750(1)**	0.452(4)	0.128(3)	**0.987(1)**	0.436(3)	**0.377(1)**	**0.365(1)**	**0.727(1)**	**2.091**

TABLE 6: The performance of multi-label approaches in terms of F1 metric.

	Emotions	Scene	Yeast	Medical	Enron	Corel5k	Tmc2007	Mediamill	NUS-WIDE	Eukaryote	VOC2007	mean rank
BR[1]	0.469(8)	0.714(6)	0.650(7)	0.328(11)	0.582(4)	0.047(6)	0.934(4)	0.557(5)	0.297(4)	0.124(9)	0.741(8)	6.545
CC[m]	0.461(10)	0.742(5)	0.657(4)	0.337(10)	0.484(10)	0.048(5)	0.939(3)	0.539(7)	**0.309(1)**	0.341(3)	0.749(6)	5.818
CLR[2]	0.465(9)	0.713(7)	0.655(5)	0.742(5)	0.600(3)	**0.293(1)**	0.933(6)	0.134(10)	0.299(3)	0.119(10)	0.748(7)	6.000
HOMER[m]	0.614(3)	0.745(4)	**0.687(1)**	0.761(3)	**0.613(1)**	0.280(2)	0.934(5)	0.579(2)	0.271(8)	0.215(7)	0.558(9)	4.091
ML-C4.5[1]	0.651(2)	0.587(10)	0.614(9)	0.768(2)	0.546(9)	0.003(9)	0.126(11)	0.054(11)	-	0.263(6)	0.532(10)	7.900
ML-*k*NN[1]	0.431(11)	0.658(9)	0.628(8)	0.560(9)	0.445(11)	0.021(7)	0.699(9)	0.570(3)	0.172(10)	0.065(11)	0.755(5)	8.454
RA*k*EL[m]	0.525(7)	0.754(2)	0.661(3)	0.704(6)	0.564(5)	0.000(11)	0.904(7)	0.471(9)	0.300(2)	0.131(8)	0.757(3)	5.727
ECC[m]	0.556(6)	**0.771(1)**	0.670(2)	0.652(7)	0.602(2)	0.001(10)	0.887(8)	0.483(8)	0.279(6)	0.306(4)	0.756(4)	5.273
RF-PCT[1]	0.611(4)	0.553(11)	0.614(10)	0.616(8)	0.552(8)	0.014(8)	0.948(2)	**0.589(1)**	0.192(9)	0.288(5)	0.429(11)	7.000
GLOCAL[2]	0.573(5)	0.673(8)	0.467(11)	0.745(4)	0.562(6)	0.183(3)	0.671(10)	0.560(4)	0.272(7)	0.385(2)	0.782(2)	5.636
BR+BN[m]	0.629	0.680	0.617	-	-	-	-	-	-	-	-	-
OURS[full]	**0.658(1)**	0.746(3)	0.651(6)	**0.778(1)**	0.559(7)	0.181(4)	**0.990(1)**	0.553(6)	0.280(5)	**0.413(1)**	**0.783(1)**	**3.273**

TABLE 7: The performance of multi-label approaches in terms of micro F1 metric.

	Emotions	Scene	Yeast	Medical	Enron	Corel5k	Tmc2007	Mediamill	NUS-WIDE	Eukaryote	VOC2007	mean rank
BR[1]	0.509(9)	0.761(4)	0.652(6)	0.343(11)	0.564(6)	0.059(5)	0.932(4)	0.533(6)	0.208(8)	0.193(9)	0.742(8)	6.909
CC[m]	0.503(10)	0.757(7)	0.650(7)	0.350(10)	0.482(10)	0.059(5)	0.936(3)	0.509(7)	0.204(9)	0.346(4)	0.743(7)	7.182
CLR[2]	0.512(8)	0.758(6)	0.655(5)	0.721(5)	0.585(3)	**0.293(1)**	0.930(5)	0.118(10)	0.213(7)	0.191(10)	0.749(6)	6.000
HOMER[m]	0.588(4)	0.764(2)	**0.673(1)**	0.773(2)	0.591(2)	0.275(2)	0.927(6)	0.553(4)	0.257(4)	0.266(7)	0.575(9)	3.909
ML-C4.5[1]	0.655(3)	0.593(11)	0.610(10)	0.756(4)	0.512(9)	0.004(9)	0.135(11)	0.007(11)	-	0.287(5)	0.536(10)	8.300
ML-*k*NN[1]	0.457(11)	0.661(10)	0.625(8)	0.634(9)	0.466(11)	0.030(7)	0.682(10)	0.545(5)	0.322(2)	0.108(11)	0.752(5)	8.091
RA*k*EL[m]	0.533(7)	**0.772(1)**	0.656(4)	0.714(6)	0.548(7)	0.000(11)	0.890(7)	0.440(9)	0.214(6)	0.202(8)	0.756(4)	6.364
ECC[m]	0.554(6)	0.762(3)	0.658(3)	0.714(6)	0.582(4)	0.002(10)	0.869(8)	0.453(8)	0.169(10)	0.368(3)	0.761(3)	5.818
RF-PCT[1]	0.672(2)	0.669(9)	0.617(9)	0.693(8)	0.537(8)	0.018(8)	0.945(2)	0.563(3)	0.221(5)	0.279(6)	0.416(11)	6.455
GLOCAL[2]	0.581(5)	0.688(8)	0.491(11)	0.762(3)	**0.593(1)**	0.198(4)	0.684(9)	**0.581(1)**	0.319(3)	0.373(2)	0.778(2)	4.455
BR+BN[m]	0.660	0.680	0.639	-	-	-	-	-	-	-	-	-
OURS[full]	**0.704(1)**	0.760(5)	0.666(2)	**0.801(1)**	0.567(5)	0.216(3)	**0.992(1)**	0.572(2)	**0.442(1)**	**0.462(1)**	**0.782(1)**	**2.091**

TABLE 8: The performance of multi-label approaches in terms of macro F1 metric.

	Emotions	Scene	Yeast	Medical	Enron	Corel5k	Tmc2007	Mediamill	NUS-WIDE	Eukaryote	VOC2007	mean rank
BR[1]	0.440(9)	0.765(5)	0.392(4)	0.361(3)	0.143(6)	0.021(5)	0.942(3)	0.056(6)	0.014(8)	0.049(10)	0.718(8)	6.091
CC[m]	0.420(10)	0.762(6)	0.390(6)	0.371(2)	0.153(4)	0.021(5)	0.947(2)	0.052(7)	0.018(6)	0.094(5)	0.721(7)	5.455
CLR[2]	0.443(8)	0.762(7)	0.392(5)	0.281(6)	0.149(5)	0.042(2)	0.938(4)	0.037(8)	0.016(7)	0.050(9)	0.726(5)	6.000
HOMER[m]	0.570(5)	0.768(3)	**0.448(1)**	0.282(5)	0.167(2)	0.036(3)	0.924(5)	0.073(4)	0.066(3)	0.093(7)	0.519(9)	4.273
ML-C4.5[1]	0.630(3)	0.596(11)	0.370(7)	0.250(7)	0.115(9)	0.008(8)	0.123(11)	0.003(11)	-	0.094(5)	0.480(10)	8.200
ML-*k*NN[1]	0.385(11)	0.692(9)	0.336(10)	0.192(11)	0.087(11)	0.010(7)	0.493(9)	0.113(2)	0.089(2)	0.023(11)	0.726(5)	8.000
RA*k*EL[m]	0.488(7)	**0.777(1)**	0.359(8)	0.210(8)	0.115(9)	0.000(11)	0.826(8)	0.019(10)	0.014(8)	0.056(8)	0.731(4)	7.455
ECC[m]	0.500(6)	0.770(2)	0.350(8)	0.203(10)	0.140(7)	0.001(10)	0.834(7)	0.022(9)	0.014(8)	0.115(2)	0.737(3)	6.000
RF-PCT[1]	0.650(2)	0.658(10)	0.322(11)	0.207(9)	0.122(8)	0.004(9)	0.857(6)	0.112(3)	0.065(4)	0.111(3)	0.335(11)	6.909
GLOCAL[2]	0.579(4)	0.699(8)	0.426(2)	**0.393(1)**	**0.172(1)**	**0.303(1)**	0.594(10)	**0.175(1)**	0.062(5)	0.111(3)	0.758(2)	3.455
OURS[full]	**0.697(1)**	0.766(4)	0.399(3)	0.342(4)	0.162(3)	0.029(4)	**0.989(1)**	0.071(5)	**0.144(1)**	**0.133(1)**	**0.762(1)**	**2.545**

order methods in most cases, as expected. For example, compared with low-order methods BR, ML-C4.5, ML-kNN, RF-PCT, and CLR, high-order method HOMER, CC, and ECC have better performances in most scenarios; compared with first-order methods BR, ML-C4.5, ML-kNN, and RF-PCT, second-order method CLR achieves better performances in most cases. First-order methods ignore label correlations, which makes it more difficult to train a single-label classifier for each label when the training data are imbalanced. Although second-order methods realize the significance of the label correlations, they only use pairwise label correlations, which are not enough to learn a satisfying multi-label classifier. However, there are always some exceptions. High-order methods RAkEL underperforms the second-order method CLR in most cases. RF-PCT is a first-order method, but on the Mediamill database it achieves the best performance in terms of accuracy and F1 and the third-best performance in terms of micro F1 and macro F1. This may be because RF-PCT is an ensemble through random forest, which has strong discriminating power [55].

Secondly, when compared to nine multi-label learning methods, the proposed method OURS achieves the best overall performance. **Following Zhu *et al.*'s work [55], we have provided the ranking of all algorithms on each database. The proposed method achieves higher ranking than any one of the compared methods on most databases, and** the "mean rank" of OURS is lowest in four metrics, which demonstrates the superiority of the proposed method that adopts an adversarial model to exploit all label correlations. The nine multi-label learning methods either ignore label correlations, exploit pairwise label correlations, or exploit correlations embedded in the subsets of all labels. Of the nine multi-label learning methods, HOMER achieves the best performance. The HOMER algorithm constructs a hierarchy of multi-label classifiers. Each classifier deals with a much smaller set of labels, which is called the meta-label [31]. At the leaf node, the classifier deals with one single label, but the number of training samples distributed to each single-label classifier may be too small for good classification. Additionally, the HOMER algorithm is appropriate for large multi-label databases. Since the meta-labels are built via balanced label clustering, obtaining good label clusters requires a many labels [55]. So the performance of the HOMER algorithm are not as good on the Emotion database, the Scene database, the Tmc2007 database, the Eukaryote database, and the VOC2007 database, as they have smaller label dimensionality.

Thirdly, compared to GLOCAL, the proposed method achieves better performance on most databases, i.e., the Emotions, Scene, Yeast, Medical, Tmc2007, NUS-WIDE, Eukaryote, and the VOC2007 databases. This demonstrates the advancement of the proposed adversarial framework. GLOCAL captures both global and local label correlations. However, GLOCAL only captures pairwise label dependencies, while the proposed method models the joint distribution of all labels, which contains all kinds of label correlations.

Fourthly, when compared to BR+BN, the proposed method OURS achieves better performances in all scenarios, demonstrating the effectiveness of the proposed method in exploiting joint label distribution. Specifically, on the Emotion database, the performances of OURS are 2.9%, 2.9%, and 4.4% higher than BR+BN in accuracy, F1, and micro F1, respectively. On the Scene database, the performances of OURS are 5.3%, 6.6%, and 8.0% higher than BR+BN in accuracy, F1, and micro F1, respectively. On the Yeast database, the performances of OURS are 2.2%, 3.4%, and 2.1% higher than BR+BN in accuracy, F1, and micro F1 respectively. BN+BR first learns the joint distribution of the ground truth label and then adjusts the results of BR according to the learned distribution. However, there are flaws when using bayesian networks for learning distribution. Some weak label correlations are neglected to limit the complexity of the network, so some nodes are not linked directly. Also, due to the Markov assumption, bayesian networks capture local and pair wised dependences among labels. Learning distribution with bayesian networks cannot exploit label correlations thoroughly and effectively. OURS introduces an adversarial model to force similarity between the distribution of the predicted labels and the distribution of the ground truth labels. Therefore, OURS can exploit label correlations more thoroughly and effectively.

TABLE 9: Comparisons between the proposed method and the RBM-based methods on the Emotions database.

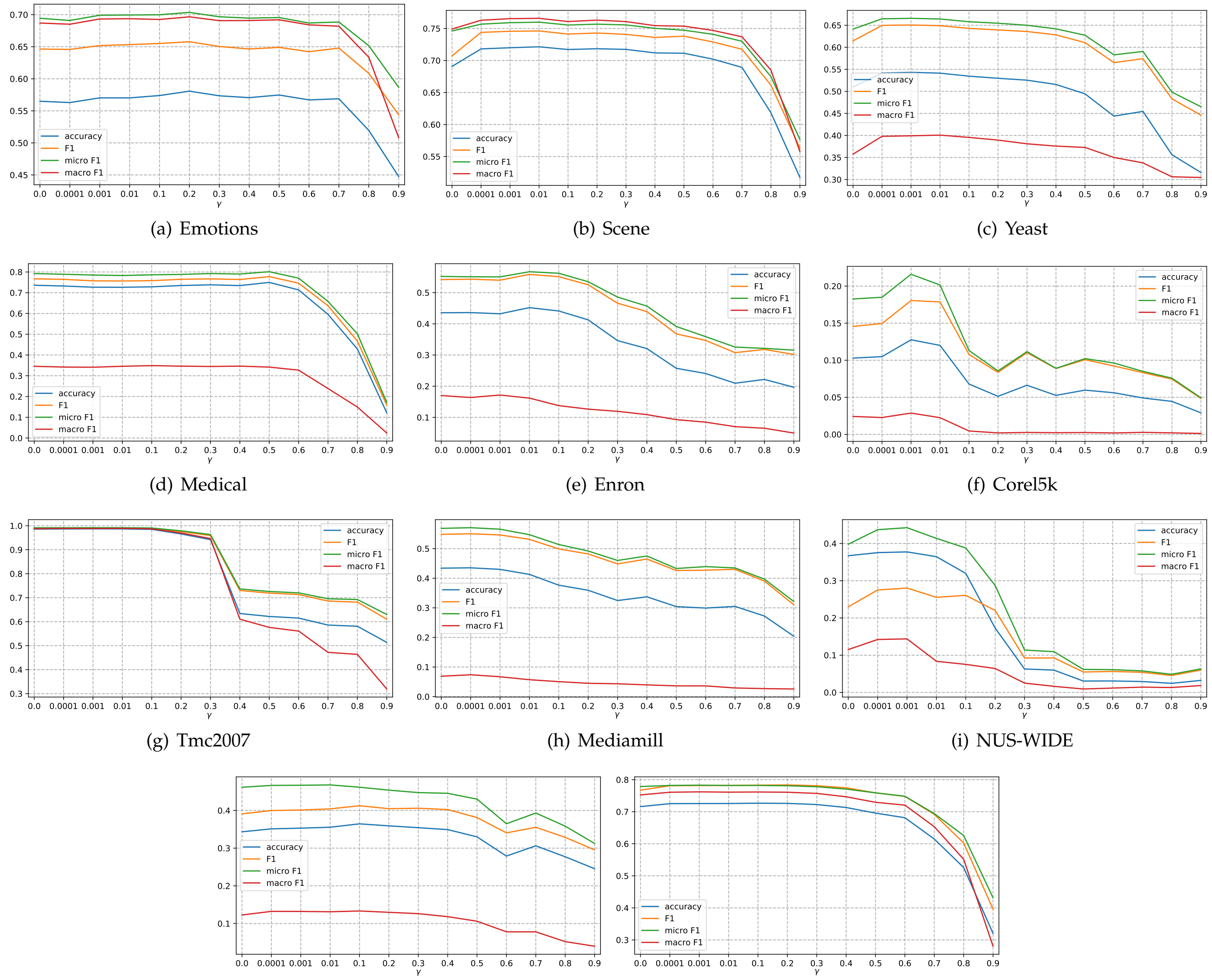
	Accuracy	F1	Micro F1	Macro F1
TRBM	0.554	0.645	0.675	0.676
FRBM	**0.585**	**0.668**	0.695	0.688
OURS	0.581	0.658	**0.704**	**0.697**

TABLE 10: Comparisons between the proposed method and CGL on the VOC2007 database.

	Accuracy	F1	Micro F1	Macro F1
CGL	0.676±0.009	0.730±0.009	0.680±0.007	0.726±0.008
OURS-V	**0.709±0.006**	**0.768±0.006**	**0.768±0.004**	**0.741±0.007**

fifthly, the comparisons between OURS and the RBM-based methods are shown in Table 9. OURS performs better that TRBM on all metrics, demonstrating the superiority of the proposed method. TRBMs consist of three layers: a measurement layer, a label layer, and a hidden layer. Measurement is the output of the basic classifier, and BR was selected as the measurement classifier in [16]. The testing samples' multiple labels are inferred by combining the measurement and the relations among multiple labels, as well as the relations between labels and measurements.

When compared to FRBM, OURS performs better on micro F1 and macro F1, but performs worse in accuracy and F1. This may be because FRBM trains with more instances. Specifically, FRBM uses a ten-fold cross-validation strategy in which nine of the ten samples are training samples. OURS uses 391 samples as a training set, which is two-thirds of the total. FRBM consists of four layers: a feature layer, first hidden layer, label layer, and second hidden layer. FRBM considers not only relations among multiple labels but also relations between features and labels. The testing samples' multiple labels are inferred from features directly, so FRBM does not need a basic classifier. Although TRBM and FRBM can capture joint label distribution for multi-label



(a) Emotions

(b) Scene

(c) Yeast

(d) Medical

(e) Enron

(f) Corel5k

(g) Tmc2007

(h) Mediamill

(i) NUS-WIDE

(j) Eukaryote

(k) VOC2007

Fig. 2: Experimental results of four metrics on eleven databases varying the γ parameter in the range from 0 to 0.9.

learning, both of them assume the form of joint distribution of all nodes and need a complex inference algorithm. The proposed method does not need any assumptions and has a simple testing process.

Lastly, on the VOC2007 database, we also compare to the state-of-the-art work, CGL. We use VGG model and conduct five-fold cross validation experiments as Li *et al.* [56] did. The comparison is shown in Table 10. The results of CGL are copied from [56]. We find that OURS-V performs better and more stable than CGL in four metrics. CGL only captures pairwise correlations by Bayesian framework, and is unable to exploit high-order label dependency. The proposed method, by contrast, exploits label distribution that contains all kinds of label dependencies, and thus achieves better performance.

5.7 The Effect of Hyper Parameter γ

In Section 5.5, we conduct experiments on OURS-W, which removes the adversarial model by setting the hyper parameter $\gamma = 0$. In this section, we investigate the impact of the hyper parameter γ by running the same set of experiments while varying the γ from 0 to 0.9, and then seek a good balance between the supervised loss and the adversarial loss in Equation 5. Fig. 2 shows the results of four metrics on eleven databases. On most databases, the performance increases slightly and then decreases as γ increases, with the rate of decline gradually increasing. This demonstrates the validity of jointly optimizing the supervised loss and the adversarial loss. When γ is very small, the supervised loss plays a major role in the parameter's update, and the adversarial model does not have as much of an impact. When γ is very large, the supervised information is not used effectively. The best hyper parameters γ on the Emotion, Scene, Yeast, Medical, Enron, Corel5k, Mediamill, NUS-WIDE, Eukaryote, and VOC2007 databases are 0.2, 0.01, 0.001, 0.5, 0.01, 0.001, 0.0001, 0.001, 0.1, and 0.1, respectively. **From the above experiments, we find the supervised loss is more important than the adversarial loss, and the optimal**

γ in all databases are no greater than 0.5. Therefore, for a new database, we recommend the selection of γ by varying γ from 0.1 to 0.5. If the performance of the proposed method decreases gradually, vary γ from 0.1 to 0.0001. We have analyzed that OURS can not achieve a significant improvement over OURS-W on the Tmc2007 database, so we can see that in Fig. 2(g), there is not an increasing phase.

6 Conclusion

In this paper, we propose a novel multi-label learning method that not only minimizes the traditional supervised loss but also leverages the joint label distribution containing all label correlations. When learning joint label distribution, we do not assume the form of the distribution. We propose an adversarial model to enforce distribution similarity between the predicted labels and the ground truth labels directly. We combine the adversarial model with the traditional supervised model and find a good balance between the two objectives by analysing the effect of trade-off parameter γ. The experimental results on eleven databases demonstrate the effectiveness of the proposed method. The ablation study demonstrates that the proposed method explicitly exploits the label correlations with the help of the adversarial model, improving the performances on multi-label classification task. Comparisons to nine multi-label algorithms, GLOCAL, BN+BR, TRBM, FRBM, and CGL, demonstrate the superiority of the proposed method in exploiting label correlations.

The "mean rank" measure we use can qualitatively represent the overall performance of methods to a certain extent, but it is not a very rigorous measure. We will consider more rigorous measure in our future work.

Acknowledgments

This work was supported in part by the National Science Foundation of China under Grant 61473270, Grant 91748129, Grant 61175037, and Grant 61228304, and in part by the Project from Anhui Science and Technology Agency under Grant 1508085SMF223.

References

[1] J. Tang, X. Shu, G.-J. Qi, Z. Li, M. Wang, S. Yan, and R. Jain, "Tri-clustered tensor completion for social-aware image tag refinement," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 8, pp. 1662–1674, 2017.

[2] X. Shu, J. Tang, G.-J. Qi, Z. Li, Y.-G. Jiang, and S. Yan, "Image classification with tailored fine-grained dictionaries," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 2, pp. 454–467, 2018.

[3] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, "Hcp: A flexible cnn framework for multi-label image classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 9, pp. 1901–1907, 2016.

[4] R. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino, "Matrix completion for weakly-supervised multi-label image classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 1, pp. 121–135, 2015.

[5] E. L. Mencía and J. Fürnkranz, "Efficient multilabel classification algorithms for large-scale problems in the legal domain," in *Semantic Processing of Legal Texts*. Springer, 2010, pp. 192–215.

[6] T. N. Rubin, A. Chambers, P. Smyth, and M. Steyvers, "Statistical topic models for multi-label document classification," *Machine learning*, vol. 88, no. 1-2, pp. 157–208, 2012.

[7] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. P. Vlahavas, "Multi-label classification of music into emotions." in *Proceedings of the 9th International Conference on Music Information Retrieval*, vol. 8, 2008, pp. 325–330.

[8] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas, "Multi-label classification of music by emotion," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2011, no. 1, p. 4, 2011.

[9] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme, "Multi-relational matrix factorization using bayesian personalized ranking for social network data," in *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 2012, pp. 173–182.

[10] S. Peters, L. Denoyer, and P. Gallinari, "Iterative annotation of multi-relational social networks," in *International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2010, pp. 96–103.

[11] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[12] Z. Barutcuoglu, R. E. Schapire, and O. G. Troyanskaya, "Hierarchical multi-label prediction of gene function," *Bioinformatics*, vol. 22, no. 7, pp. 830–836, 2006.

[13] M.-L. Zhang and K. Zhang, "Multi-label learning by exploiting label dependency," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 999–1008.

[14] S. Wang, J. Wang, Z. Wang, and Q. Ji, "Enhancing multi-label classification by modeling dependencies among labels," *Pattern Recognition*, vol. 47, no. 10, pp. 3405–3413, 2014.

[15] Z. Wang, Y. Li, S. Wang, and Q. Ji, "Capturing global semantic relationships for facial action unit recognition," in *International Conference on Computer Vision (ICCV)*. IEEE, 2013, pp. 3304–3311.

[16] S. Wang, J. Wang, Z. Wang, and Q. Ji, "Multiple emotion tagging for multimedia data by exploiting high-order dependencies among emotions," *IEEE Transactions on Multimedia*, vol. 17, no. 12, pp. 2185–2197, 2015.

[17] S. Wu, S. Wang, and Q. Ji, "Capturing dependencies among labels and features for multiple emotion tagging of multimedia data." in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017, pp. 1026–1033.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[19] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[20] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[21] A. Clare and R. D. King, "Knowledge discovery in multi-label phenotype data," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2001, pp. 42–53.

[22] H. Blockeel, L. De Raedt, and J. Ramon, "Top-down induction of clustering trees," in *Proceedings of the 15th International Conference on Machine Learning*. Morgan Kaufmann, 1998, pp. 55–63.

[23] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker, "Label ranking by learning pairwise preferences," *Artificial Intelligence*, vol. 172, no. 16-17, pp. 1897–1916, 2008.

[24] J. Fürnkranz, E. Hüllermeier, E. L. Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Machine learning*, vol. 73, no. 2, pp. 133–153, 2008.

[25] S.-H. Park and J. Fürnkranz, "Efficient pairwise classification," in *European Conference on Machine Learning*. Springer, 2007, pp. 658–665.

[26] E. L. Mencía, S.-H. Park, and J. Fürnkranz, "Efficient voting prediction for pairwise multilabel classification," *Neurocomputing*, vol. 73, no. 7-9, pp. 1164–1176, 2010.

[27] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Advances in neural information processing systems*, 2002, pp. 681–687.

[28] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, 2006.

[29] N. Ghamrawi and A. McCallum, "Collective multi-label classification," in *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, 2005, pp. 195–200.

[30] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," in *European conference on machine learning*. Springer, 2007, pp. 406–417.

[31] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Effective and efficient multilabel classification in domains with large number of labels," in *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD08)*, vol. 21. sn, 2008, pp. 53–59.

[32] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine learning*, vol. 85, no. 3, p. 333, 2011.

[33] D. Kocev, C. Vens, J. Struyf, and S. Džeroski, "Ensembles of multi-objective decision trees," in *European Conference on Machine Learning*. Springer, 2007, pp. 624–631.

[34] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Džeroski, "An extensive experimental comparison of methods for multi-label learning," *Pattern recognition*, vol. 45, no. 9, pp. 3084–3104, 2012.

[35] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, 2007.

[36] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data mining and knowledge discovery handbook*. Springer, 2009, pp. 667–685.

[37] M. S. Sorower, "A literature survey on algorithms for multi-label learning," *Oregon State University, Corvallis*, vol. 18, pp. 1–25, 2010.

[38] E. Gibaja and S. Ventura, "Multi-label learning: a review of the state of the art and ongoing research," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 4, no. 6, pp. 411–444, 2014.

[39] X. Geng, "Label distribution learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1734–1748, 2016.

[40] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[41] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[42] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: introduction and outlook," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 588–598, 2017.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[44] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning." in *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*, 2016, pp. 265–283.

[45] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, "Mulan: A java library for multi-label learning," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2411–2414, 2011.

[46] P. Duygulu, K. Barnard, J. F. de Freitas, and D. A. Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary," in *European conference on computer vision*. Springer, 2002, pp. 97–112.

[47] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "Nuswide: a real-world web image database from national university of singapore," in *Proceedings of the ACM international conference on image and video retrieval*. ACM, 2009, pp. 48:1–48:9.

[48] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[49] E. Spyromitros-Xioufis, S. Papadopoulos, I. Y. Kompatsiaris, G. Tsoumakas, and I. Vlahavas, "A comprehensive study over vlad and product quantization in large-scale image retrieval," *IEEE Transactions on Multimedia*, vol. 16, no. 6, pp. 1713–1728, 2014.

[50] B. Klimt and Y. Yang, "The enron corpus: A new dataset for email classification research," in *European Conference on Machine Learning*. Springer, 2004, pp. 217–226.

[51] J. P. Pestian, C. Brew, P. Matykiewicz, D. J. Hovermale, N. Johnson, K. B. Cohen, and W. Duch, "A shared task involving multi-label classification of clinical free text," in *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*. Association for Computational Linguistics, 2007, pp. 97–104.

[52] A. N. Srivastava and B. Zane-Ulman, "Discovering recurring anomalies in text reports regarding complex space systems," in *Aerospace conference*. IEEE, 2005, pp. 3853–3862.

[53] J. Xu, J. Liu, J. Yin, and C. Sun, "A multi-label feature extraction algorithm via maximizing feature variance and feature-label dependence simultaneously," *Knowledge-Based Systems*, vol. 98, pp. 172–184, 2016.

[54] C. G. Snoek, M. Worring, J. C. Van Gemert, J.-M. Geusebroek, and A. W. Smeulders, "The challenge problem for automated detection of 101 semantic concepts in multimedia," in *Proceedings of the 14th ACM international conference on Multimedia*. ACM, 2006, pp. 421–430.

[55] Y. Zhu, J. T. Kwok, and Z.-H. Zhou, "Multi-label learning with global and local label correlation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1081–1094, 2018.

[56] Q. Li, M. Qiao, W. Bian, and D. Tao, "Conditional graphical lasso for multi-label image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2977–2986.

Shangfei Wang received her BS in Electronic Engineering from Anhui University, Hefei, Anhui, China, in 1996. She received her MS in circuits and systems, and the PhD in signal and information processing from University of Science and Technology of China (USTC), Hefei, Anhui, China, in 1999 and 2002. From 2004 to 2005, she was a postdoctoral research fellow in Kyushu University, Japan. Between 2011 and 2012, Dr. Wang was a visiting scholar at Rensselaer Polytechnic Institute in Troy, NY, USA. She is currently an Associate Professor of School of Computer Science and Technology, USTC. Her research interests cover affective computing and probabilistic graphical models. She has authored or co-authored over 90 publications. She is a senior member of the IEEE and a member of the ACM.

Guozhu Peng received his BS in mathematics from South China University of Technology in 2016, and he is currently pursuing his MS in Computer Science in the University of Science and Technology of China, Hefei, China. His research interesting is affective computing.

Zhangqiang Zheng received his BS in mathematics from Liaoning Technical University in 2017, and he is currently pursuing his MS in Computer Science in the University of Science and Technology of China, Hefei, China. His research interesting is affective computing.