Multilabel Consensus Classification

Sihong Xie[†] Xiangnan Kong[†] Jing Gao[§] Wei Fan[‡] Philip S. Yu[†]

***Abstract*— In the era of big data, a large amount of noisy and incomplete data can be collected from multiple sources for prediction tasks. Combining multiple models or data sources helps to counteract the effects of low data quality and the bias of any single model or data source, and thus can improve the robustness and the performance of predictive models. Out of privacy, storage and bandwidth considerations, in certain circumstances one has to combine the predictions from multiple models or data sources without accessing the raw data. Consensus-based prediction combination algorithms are effective for such situations. However, current research on prediction combination focuses on the single label setting, where an instance can have one and only one label. Nonetheless, data nowadays are usually multilabeled, such that more than one label have to be predicted at the same time. Direct applications of existing prediction combination methods to multilabel settings can lead to degenerated performance. In this paper, we address the challenges of combining predictions from multiple multilabel classifiers and propose two novel algorithms, MLCM-r (MultiLabel Consensus Maximization for ranking) and MLCM-a (MLCM for microAUC). These algorithms can capture label correlations that are common in multilabel classifications, and optimize corresponding performance metrics. Experimental results on popular multilabel classification tasks verify the theoretical analysis and effectiveness of the proposed methods.**

I. Introduction

Combining multiple models or data sources has been attracting more and more attentions in data mining and machine learning research communities. Real-world data are usually massive, noisy and incomplete. To improve the robustness and generalization ability of learning methods on these real-world data, one has to combine multiple models and exploit the knowledge of multiple data sources. Many methods have been proposed for the purpose, such as [7, 1], which focus on learning ensembles of models from the training data and predictions on test data. Due to privacy, bandwidth or storage issues, there are situations where we cannot have access to either the training data nor the testing data directly. Instead, only the predictions of base models are available. For example, in finance, aggregating customers information from multiple banks would benefit customer segmentation analysis. However, it would be unsafe or infeasible to transfer the customer information across different banks. One solution to this problem is that we can apply the analysis at each bank individually, and then aggregate the predictions from multiple banks. Prediction combination is a powerful paradigm for such situations with an abundance of studies [4, 5, 6, 10]. These algorithms combine the predictions of multiple supervised and/or unsupervised models, in hope of improving accuracies by exploiting the strengths of different models or data sources, without access to training or test data.

Conventional research on prediction combination has been focusing on single label classification and cannot handle multilabel classification. Meanwhile, multilabel classification has seen its wide application in text/image categorization, bioinformatics and so on, and therefore is of practical importance. Although certain ensemble methods [9, 15] have been proposed to handle multilabel classification, they focus on building the ensemble from training data, not on prediction combination. Given the practical needs to combine multilabel predictions from multiple models/data sources without training and test data, we identify the following challenges that need to be addressed in order to bridge the gap. First, although state-of-the-art multilabel classification methods show that label correlations can help improving classification performances, how to exploit label correlations solely using predictions of base models has not been addressed before. Second, there are various evaluation metrics for multilabel classification, such as microAUC, ranking loss, one error, etc. [3], it is more desirable to design algorithms that can be proved to be optimal for a specific metric, as different applications require different quality measures. It is non-trivial to align prediction combination with the modeling of label correlation to optimize a specific metric. There is no existing work that addresses the above issues.

In this paper, we address the above challenges by proposing two different algorithms that can model label correlations given only the predictions of base models. The algorithms are designed and proved to optimize two widely used but fundamentally different evaluation metrics, respectively. The first algorithm MLCM-r consolidates the predictions of base models via maximizing model consensus and exploits label correlations using random walk in the label space. The algorithm is proved to optimize ranking loss, which measures the quality of the predictions on a per instance basis (e.g. find relevant labels for a query in image search engine). Another important multilabel performance metric is microAUC, which is different from ranking loss (Section IV). Since a model that optimizes ranking loss might not be optimal on microAUC,we propose a second algorithm

[†]Department of Computer Science, University of Illinois at Chicago
[§]Department of Computer Science and Engineering, University at Buffalo
[‡]Huawei Noah's Ark Lab, Hong Kong

Table I: Notations

Symbol	Meaning
m	Number of multilabel classifiers
n	Number of instances
l	Number of labels
$\mathbf{x}$	An instance
$\mathbf{z}$	Ground truth labels of $\mathbf{x}$
Y^k	output of the kth model
$\bar{Y}$	Average of $Y^k, k = 1, \ldots, m$
$\mathbf{y}_i^k$	prediction of the kth model for the ith instance
Y	Consolidated prediction of $Y^k, k = 1, \ldots, m$
$\langle \cdot, \cdot \rangle$	inner product of two vectors
$\lvert \cdot \rvert$	Determinant of a matrix
$\lVert \cdot \rVert$	Frobenius norm of a matrix
$\mathbb{1}[\cdot]$	indicator of a predicate
$\mathrm{card}(A)$	cardinality of the set A

called MLCM-a (MultiLabel Consensus Maximization for microAUC). MLCM-a is formulated as a optimization problem that regularizes prediction consolidation using partial correlations between labels, and we show that the objective of this formulation optimizes microAUC. The contributions of this paper can be summarized as follows.

- We first study the problem of how to combine predictions of multiple models in multilabel learning without access to training and test data.
- We propose two novel algorithms that can jointly model correlations among different labels and the consensus among multiple models. We prove that the two algorithms optimize two multilabel classification metrics, respectively. As far as we know, this is the first work that addresses the multilabel-label consensus learning problem and optimize specific metrics.
- We compare the proposed models to 3 baselines on 6 multilabel classification tasks, with a maximum of 45% percent of reduction in ranking loss and 20% percent of increase in microAUC.

II. Preliminary

In this section we recapitulate model combination and multilabel classification algorithms, along with the challenges that we are addressing. Table I summarizes most of the symbols and their definitions used in this paper. We use boldface lower-case letters for vectors (e.g., $\mathbf{x}$) and capital letters for matrices (e.g., Y).

A. Multilabel Classification

In multilabel-label classification problems, the data are in the form of $(\mathbf{x}, \mathbf{z})$, where $\mathbf{x}$ is the feature vector of an instance and $\mathbf{z}$ is the label vector. Suppose L is the set of all ℓ possible labels, then $\mathbf{z}$ is a vector with length $|L| = l$ and $z_\ell \in \{0, 1\}$ denotes the value of the ℓ-th label. Multilabel classification is different from multiclass classification. In multiclass classification, an instance have only one label, which can take more than two values (or classes). However, in multilabel classification, an instance can have more than one label, each of which can take one and only one of the multiple values (classes). For example, an account on a social network (LinkedIn, Facebook, etc.) can have multiple labels such as "sex" and "is employed", while there can be only one specific value for the label "is employed". Multilabel classification introduces various unique challenges, such as sparsity and imbalance of labels, multiple performance metrics of a model, etc. Among these challenges, how to model and exploit label relationships to improve accuracy has been studied intensively in [12, 14]. There are various types of label relationships, the simplest one is pair-wise correlation, which specifies how often two labels co-occur. Recently, certain types of label relationship is shown to be connected to certain corresponding evaluation metrics. For example, it is shown in [3] that if one can compute the relevance score of each individual label given an instance, the ranking according to the scores would yield the minimum ranking loss.

B. Prediction Combination Algorithms

Given the predictions of multiple models, one needs to combine the predictions in order to obtain a single final prediction. Suppose there are m base models, whose predictions can be denoted by $\{Y^1, \ldots, Y^m\}$. For $k = 1, \ldots, m$, Y^k is an $n \times l$ matrix, the (i, ℓ) element $Y_{i\ell}$ gives the class value of the i-th instance for the ℓ-th label, according to the k-th model. Y^k is a binary matrix specifying the presence of a label in an instance.

Simple averaging The simplest way to consolidate predictions from multiple models is to take the average of the predictions [5]:

$$Y = \bar{Y} = \frac{1}{m} \sum_{k=1}^{m} Y^k = \sum_{k=1}^{m} Y^k (m I_l)^{-1} \tag{1}$$

where I_l is the l dimensional identity matrix. The loss function that simple averaging minimizes is the sum of squared error between the consolidated prediction Y and the base models' predictions $\{Y^1, \ldots, Y^m\}$

$$\sum_{k=1}^{m} \|Y^k - Y\|^2 = \sum_{i=1}^{n} \sum_{k=1}^{m} \|\mathbf{y}_i^k - \mathbf{y}_i\|^2 \tag{2}$$

Simple averaging has been applied to multilabel classification to combine multiple models, such as Rakel [12] and BoosTexter [7]. These methods only model label dependencies in the training phase, and in the combination step, without access to the training and test data, they cannot handle any label dependency.

BGCM BGCM [4] is proposed to solve model combination problems for multi-class problem. Here we first introduce BGCM, based on which we propose a model combination method that also accounts for label relationships in the next section. In general, given the predictions of m classifiers for n instances, with c classes from a single label,

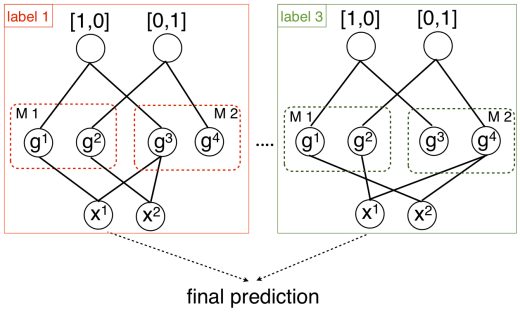
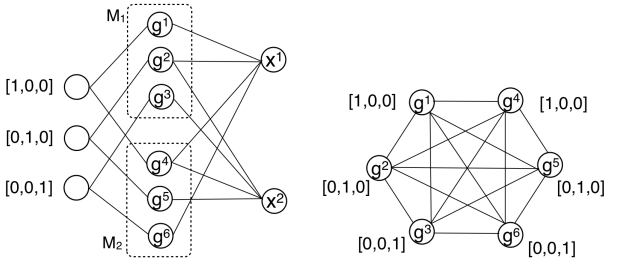


Figure 1: Applying BGCM to multilabel prediction combination



(a) MLCM-r (b) Graph of group nodes encoding label relationships

Figure 2: Bipartite graph for MLCM-r and its collapse to group nodes

BGCM constructs a bipartite graph with n instances nodes and $v = m \times c$ group nodes. An example of the bipartite graph with 2 instances and 2 classes for a label is shown in the left rectangle in Figure 1, where group nodes are annotated with the letter g and instance nodes with the letter x. Each node is associated with a probability distribution over c classes. The distribution for the i-th instance node is given by the row vector $\mathbf{u}_i, i = 1, \ldots, n$, which are collectively denoted by the $n \times c$ matrix $U = [\mathbf{u}'_1, \ldots, \mathbf{u}'_n]'$. Similarly, let the $v \times c$ matrix $Q = [\mathbf{q}'_1, \ldots, \mathbf{q}'_v]'$ be the distributions of v group nodes. The connections of these nodes are determined by the predictions of the base models. If $\mathbf{x}_i$ is classified into the j-th class by k-th model, the i-th instance node is connected to the $(k-1)\times c+j$ group nodes. In the above bipartite graph, instance $\mathbf{x}_1$ is classified into class 1 by model 1, so the first instance node is connected to g^1. A group node is connected to a class node to specify the class distribution of the node. In general, if a group node represents the j-th class, then it is connected to a class node with class distribution $\mathbf{b}_j$, which has 1 at its j-th position and 0 otherwise. Let the $v \times c$ matrix $B = [\mathbf{b}'_1, \ldots, \mathbf{b}'_v]'$.

For each label, BGCM solves the following optimization problem to achieve maximal consensus among base models,

$$\min_{U,Q} \quad \sum_{i=1}^{n}\sum_{j=1}^{v} a_{ij}\|\mathbf{u}_i - \mathbf{q}_j\|^2 + \alpha \sum_{j=1}^{v} \|\mathbf{q}_j - \mathbf{b}_j\|^2 \qquad (3)$$

$$\text{s.t.} \quad u_{i\ell} \geq 0, \textstyle\sum_{\ell=1}^{c} u_{i\ell} = 1, i = 1, \ldots, n \qquad (4)$$

$$q_{j\ell} \geq 0, \textstyle\sum_{\ell=1}^{c} q_{j\ell} = 1, j = 1, \ldots, v$$

In Eq.(3), $a_{ij} = 1$ indicates that the i-th instance node and the j-th group node are connected, otherwise $a_{ij} = 0$. After the optimization problem is solved, the consolidated prediction of the i-th instance for a single label can be obtained by taking the maximal value in $\mathbf{u}_i$. [6, 5, 10]. BGCM can only combine multilabel predictions by first combine the predictions for each single label, and then concatenate the predictions for individual labels to obtain the final prediction for multiple labels, as shown in Figure 1. This process treats labels independently without exploiting label correlations, and is similar to the Binary Relevance (BR) method in multilabel classification literature. Apparently, no label correlation is modeled in this paradigm. Next we propose a novel method to incorporate label correlations in multilabel predictions combination.

III. Multilabel Consensus Maximization for Ranking Loss

We propose MLCM-r, which adopts the architecture of BGCM to achieve this goal. For simplicity, we assume that each label consists of two classes. We abuse the notations introduced in Section II-B. In particular, we let the n by v ($v = m \times l$) connection matrix A encode the multilabel predictions, where the $(i, (k-1) \times l + j)$-th entry is 1 if the k-th model predicts that the i-th instance takes class 1 on the j-th label, otherwise the entry is 0. Viewing A as a connection matrix between instances and labels, a bipartite graph can be constructed for MLCM-r. An example of the bipartite graph of MLCM-r for 2 instances, 3 classes and two base multilabel classifiers is shown in Figure 2(a). Similar to the bipartite graph, the bipartite graph for MLCM-r has both group nodes and instance nodes, annotated by the letters g and x. However, there are some differences between two bipartite graphs. Surrounded by a rectangle with dashed line are the group nodes from a classifier (e.g. the rectangle M_1 includes the group nodes from the first classifier). A group node in Figure 2(a) represents a label instead of a class in Figure 1. An instance node in Figure 2(a) can be connected to more than one group nodes from a classifier, naturally representing the multilabel predictions. These differences between Figure 2(a) and Figure 1 bring more expressive power to MLCM-r, as summarized below:

- The connections between an instance and *all* labels are given by a single graph in MLCM-r, instead of being broken down into multiple bipartite graphs in BGCM.
- most importantly, the relationship between labels can be derived in MLCM-r using Figure 2(a), as shown in the graph of group nodes in Figure 2(b). We give more details of this property of MLCM-r in [13].

According to the newly defined A, we re-define the distributions associated with the nodes. $u_{i\ell}$ (the ℓ-th entry of $\mathbf{u}_i$) is now defined to be the probability of the i-th instance

taking class 1 on the ℓ-th label. Similarly $q_{j\ell}$ is defined as the probability of seeing the ℓ-th label given the j-th label (the reason of this definition is explained in the next section). If the j-th group node represents the ℓ-th label, it is connected to a label node with distribution $\mathbf{b}_j$, which has 1 on its ℓ-th entry and 0 for the other entries, let $B = [\mathbf{b}'_1, \ldots, \mathbf{b}'_v]'$ similarly as in BGCM. With the re-defined variables and constants, MLCM-r maximizes model consensus by solving a similar optimization problem in Eq.(3). For the closed form solutions for the problem and the analysis of why the solution minimizes ranking loss, please refer to the full version [13] for details.

IV. Multilabel Consensus Maximization for microAUC

A. microAUC and its properties

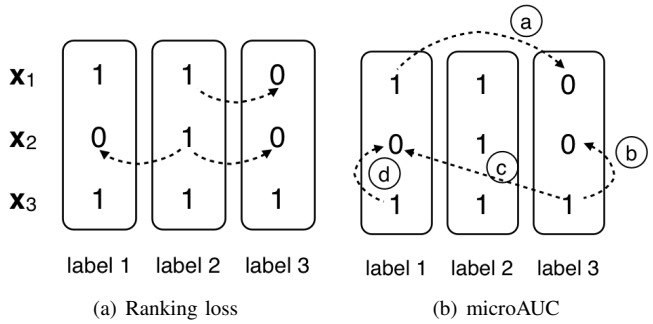
AUC (Area Under the Curve) is a binary classification metric for classification problems with skew class distributions. In multilabel classification, an instance usually has only a small number of relevant labels out of many labels. In other words, relevant labels are dominated by irrelevant labels. Thus AUC can be adopted as a metric (called "microAUC") for multilabel classification. Formally, the label matrix $Z = [\mathbf{z}'_1, \ldots, \mathbf{z}'_n]'$ for n instances has a total of $n \times l$ entries. Let P be the set of positive (relevant) entries and N the set of negative (irrelevant) entries, $\mathrm{card}(P) \ll \mathrm{card}(N)$. Given a list of relevance scores $f(\cdot)$ of all entries, microAUC [2] is defined as

$$\text{microAUC} = \sum_{i \in P} \sum_{j \in N} \frac{\mathbb{1}[f(i) > f(j)]}{\mathrm{card}(P) \times \mathrm{card}(N)} \tag{5}$$

where $f(i)$ is the relevance score of entry i. A fundamental difference between two metrics is that, ranking loss does not compare the ranks between labels of two different instances, while microAUC compares the ranks of all possible pairs of labels, no matter they are from the same instance or not. This difference is demonstrated in Figure 3, where the ground truth labels of 3 instances $\{\mathbf{x}_1, \ldots, \mathbf{x}_3\}$ with 3 labels are given in rectangles. The entries are grouped in rectangles according to the labels, while each row represents the labels of an instance. We use arrows in Figure 3(a) and 3(b) to indicate pairs of labels that are accounted for by ranking loss and microAUC, respectively. In Figure 3(b), arrow **a** is a pair of entries considered by ranking loss, arrow **b** indicates pairs of entries within a label for different instances, and arrow **c** points from a label of an instance to a different label of a different instance.

B. MLCM-a

In this section, we motivate and propose a novel algorithm to model label relationships when combining predictions to optimize microAUC. Continuing the above example. Pairs of entries indicated by arrow **b** or **d** must have been handled by any reasonable base models, which predict the relevance of



(a) Ranking loss (b) microAUC

Figure 3: Comparison of ranking loss and microAUC

a label to the instances. Pairs of entries indicated by arrow **a** consist only a small portion of all pairs if n is large, due to the sparsity of relevant labels. Therefore, the major challenge in optimizing microAUC is how to enforce preference of one label over other labels across different instances, such as the pairs indicated by arrow **c**. For the two instances $\mathbf{x}_2$ and $\mathbf{x}_3$, we need to estimate the posteriors $p(y_j = 1|\mathbf{x}_i)$ for $i \in \{2, 3\}, j \in \{1, 3\}$, in order to derive the relevance to account for the pair by arrow **c**. We show that we can exploit the label correlations to guarantee that there is no error incurred due to the pair of labels connected by arrow **c**, with high probability. Assume that we have known, with high probability, that $p(y_1 = 1|\mathbf{x}_3) > p(y_1 = 1|\mathbf{x}_2)$ (arrow **d**). Also assume that with high probability, label 1 and 3 are correlated. Then to avoid incurring an error due to $p(y_1 = 1|\mathbf{x}_2) > p(y_3 = 1|\mathbf{x}_3)$ (arrow **c**), we simply enforce the estimation of $p(y_3 = 1|\mathbf{x}_2)$ and $p(y_3 = 1|\mathbf{x}_3)$ to match those of $p(y_1 = 1|\mathbf{x}_2)$ and $p(y_1 = 1|\mathbf{x}_3)$, namely, $p(y_3 = 1|\mathbf{x}_2) < p(y_3 = 1|\mathbf{x}_3)$ (arrow **b**). Then we have $p(y_3 = 1|\mathbf{x}_3) > p(y_1 = 1|\mathbf{x}_2)$ (arrow **c**) with certain high probability.

In summary, we can tackle the challenge in two steps:

- estimate the correlations between labels accurately
- estimating the consolidated label relevance to optimize microAUC using the estimated label correlations.

Regarding the second step, we model correlation between a pairs of label using partial correlation matrix. Specifically, partial correlation between labels ℓ and ℓ' is the correlation between the two labels given the other labels, and the partial correlation matrix of labels is given by the inverse of the label correlation matrix Ω, which is assumed to be known. Given $Y_1, \ldots, Y_k$, we set up an optimization objective with two goals to obtain the consolidated labels Y that is close to Ω^{-1}. The first goal is to minimize certain consensus loss function employed by model combination algorithms, here we choose the loss function used by simple averaging (Eq.(2)). The second goal is to maximize the correlation between the label partial correlation and the empirical label correlation, namely, $\mathrm{tr}(Y'Y\Omega^{-1}) = \mathrm{tr}(Y\Omega^{-1}Y')$. The optimization problem is then given by

$$\min_{Y} \quad J = \|\bar{Y} - Y\|^2 + \mathrm{tr}(Y\Omega^{-1}Y') \tag{6}$$

Table II: Datasets

datasets	# of instances	# of features	# of labels
enron	1702	1054	53
medical	978	1449	45
rcv1 subset 1	2997	47337	101
rcv1 subset 2	2951	47337	101
slashdot	3782	1101	22
bibtex	3701	1995	159

where Y is the consolidated labels. The optimal solution is

$$\mathbf{y}_i = \sum_{k=1}^{m} \mathbf{y}_i^k (\Omega^{-1} + mI_l)^{-1} = m\bar{\mathbf{y}}_i(\Omega^{-1} + mI_l)^{-1} \quad (7)$$

Note that we assume Ω is given in the above optimization problem. In reality, Ω is usually unknown and has to be estimated from data. If we assume $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ are the data independently generated from the normal density $\mathbf{y}_i \sim \mathcal{N}(\mathbf{0}, \Omega)$, then Ω is estimated as $\hat{\Omega}_{MLE} = \frac{1}{n} Y'Y$ Now we can put the above two steps together to build the MLCM-a algorithm, as described in Algorithm 1.

Algorithm 1 MLCM-a

1: **Input**: Predictions from base models $\{Y^1, \ldots, Y^m\}$
2: **Output**: Consolidated predictions Y.
3: Estimate $Y = \bar{Y}$
4: **for** $t = 1 \to T$ **do**
5: Estimate covariance $\Omega = \frac{1}{n} Y'Y$
6: Estimate Y using Eq.(7)
7: **end for**

V. Experiments

A. Experimental Settings

Datasets With 6 datasets widely used in multilabel classification community (see Table II), we demonstrate the effectiveness of the proposed methods.

Evaluation Metrics Besides ranking loss and microAUC, we further include "one error" and "average precision" to give some empirical observations of the proposed methods, please refer to [14] for the details of these metrics.

Baselines A base model is obtained by first randomly shuffling the dataset, followed by 10-fold CV to obtain predictions for all instances. We repeat this process 10 times to obtain 10 base models. The averaged performance of these base models (denoted by BM in the sequel) are treated as one of the baselines. The predictions of these base models are used as input to majority voting (MV in the sequel), MLCM-r and MLCM-a, each of which produces consolidated predictions, based on which we can evaluate the performance of MV, MLCM-r and MLCM-a. This experiment is repeated for 10 times for each dataset and the averaged performance is reported.

Table III: Results on enron dataset

Methods	Metrics			
	microAUC	one error	ranking loss	avg precision
BM	0.7342	0.5024	0.2967	0.4592
MV	0.8289	0.3398	0.1848	0.6020
MLCM-r	0.8759	0.6233	**0.1003**	0.5252
MLCM-a	**0.8931**	**0.2675**	0.1070	**0.6556**

Table IV: Results on medical dataset

Methods	Metrics			
	microAUC	one error	ranking loss	avg precision
BM	0.8887	0.2041	0.0989	0.7953
MV	0.9321	0.1410	0.0582	0.8639
MLCM-r	0.9536	0.1327	**0.0494**	**0.8750**
MLCM-a	**0.9556**	**0.1322**	0.0530	0.8649

B. Results

We show the performance of the proposed algorithms and baselines in Table III-VIII. We have a couple of observations. First, by comparing results in the rows for BM and MV, one can see that combining model can boost the performance of multilabel classification, even only using the simplest way of combination (simple averaging here). The maximum improvements of MV over BM are 41% and 12.8% for ranking loss and microAUC, respectively. This is not surprising, as this method is widely used in ensemble multilabel classification methods like [12, 9, 15]. Second, by comparing the results of the proposed methods and simple averaging, we observe that simple averaging is not sufficient to fully exploit label correlations, especially when the base models do not take the correlations into account. The maximum improvement of either the proposed algorithms over MV is 45% in ranking loss and 20% in microAUC. Third, out of 6 tasks, MLCM-r wins MLCM-a 5 times in ranking loss, with a maximum of 12% improvement, and MLCM-a wins MLCM-r 4 times in microAUC, with a maximum of 5.8% improvement. The above comparisons show the superiority of the proposed methods over the baselines for multilabel predictions combination tasks, and also how to choose from the proposed methods when different metrics are considered. Lastly, besides ranking loss and microAUC, the proposed methods also outperform the baselines with the other two metrics, and this shows the wide applicability of the proposed methods.

VI. Related Work

Multilabel classification methods can be roughly categorized as following. (1) Binary Relevance. Labels are treated

Table V: Results on rcv1 subset 1 dataset

Methods	Metrics			
	microAUC	one error	ranking loss	avg precision
BM	0.6194	0.6036	0.3373	0.3218
MV	0.6787	0.4792	0.2838	0.4164
MLCM-r	0.7867	0.3554	**0.2316**	**0.5017**
MLCM-a	**0.8069**	**0.3120**	0.2605	0.4967

Table VI: Results on rcv1 subset 2 dataset

Methods	Metrics			
	microAUC	one error	ranking loss	avg precision
BM	0.6220	0.5652	0.5652	0.3659
MV	0.6678	0.4730	0.4730	0.4389
MLCM-r	0.7581	0.2955	0.2955	**0.5146**
MLCM-a	**0.8020**	**0.2830**	**0.2830**	0.5073

Table VII: Results on slashdot dataset

Methods	Metrics			
	microAUC	one error	ranking loss	avg precision
BM	0.7377	0.4875	0.2062	0.5856
MV	0.8210	0.4085	0.1482	0.6689
MLCM-r	**0.8782**	0.4123	**0.1203**	0.6736
MLCM-a	0.8702	**0.3887**	0.1289	**0.6800**

as independent and prediction of each label is handled by individual binary/multiclass model. The binary relevance paradigm does not consider label dependency and thus might be inferior to methods that consider label dependency. (2) Pairwise relationship. This category of methods model the relationships between two labels. In [14], they propose a method to learn label relationships using Bayesian network, which is later utilized to learn a binary classifiers for each label given that label's parent labels. (3) Powerset Methods [12].. This set of methods try to fully consider all possible co-occurrence of labels.

There have been an extensive study of ensemble methods, which combines the knowledge of multiple models to improve performance. The simplest ensemble method is majority voting. In [1], bootstrap sampling is used to create multiple copies of training data to derive an ensemble of models. It is shown that bagging improves performance via reduction in variance. Another famous ensemble method is boosting [7], which builds the ensemble via sequential training of base models to exploit model correlation.

Predictions combination has been researched for at least a decade. In [10], they present three methods, CSPA, HGPA and MCLA for cluster ensemble. In [4], they propose BGCM, which maximizes the consensus among models. None of these methods can directly address to the multilabel prediction combination problem.

In [11], they treat the learning of a model for a label as a stand-along task. Then their algorithm learns a linear combination of multiple kernels for each task. Their method and that proposed in [15, 9] assume that training and test data are available and therefore cannot address the challenge of this paper.

Table VIII: Results on bibtex dataset

Methods	Metrics			
	microAUC	one error	ranking loss	avg precision
BM	0.6620	0.5469	0.3095	0.3575
MV	0.7266	0.4329	0.2508	0.4567
MLCM-r	**0.8668**	0.4713	**0.1599**	0.4828
MLCM-a	0.8645	**0.3790**	0.1755	**0.4937**

VII. Conclusion

In this paper, we aim at combining multilabel predictions from multiple models. The challenge is how to exploit label correlations to optimize a certain performance metric when consolidating predictions. Existing multilabel ensemble algorithms fail to do so. We address the challenge via two methods: MLCM-r and MLCM-a. The former uses random walk in the label space to explicitly infer label correlation, which in turn results in consolidated multilabel predictions optimized for ranking loss. The latter uses an optimization framework to estimate the partial label correlations, which regularizes predictions consolidation to optimize microAUC. We analyze both algorithms to establish these optimal properties. Experimental results affirmatively demonstrate the superiority of the proposed algorithms.

***Acknowledgements*—This work is supported in part by NSF through grants CNS-1115234, DBI-0960443, and OISE-1129076, US Department of Army through grant W911NF-12-1-0066, and Huawei Grant**

References

[1] Leo Breiman. Bagging predictors. *Mach. Learn.*, 1996.
[2] Corinna Cortes and Mehryar Mohri. Auc optimization vs. error rate minimization. In *NIPS*, 2003.
[3] Krzysztof Dembczyński, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. On label dependence and loss minimization in multi-label classification. 2012.
[4] Jing Gao, Feng Liang, Wei Fan, Yizhou Sun, and Jiawei Han. Graph-based consensus maximization among multiple supervised and unsupervised models. In *NIPS*, 2009.
[5] Tao Li and Chris Ding. Weighted Consensus Clustering. SDM, 2008.
[6] Tao Li, Chris Ding, and Michael I. Jordan. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. ICDM, 2007.
[7] Robert E. Schapire. The boosting approach to machine learning: An overview. 2002.
[8] Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based systemfor text categorization. 2000.
[9] Chuan Shi, Xiangnan Kong, Philip S Yu, and Bai Wang. Multi-label ensemble learning. ECML/PKDD, 2011.
[10] Alexander Strehl and Joydeep Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *JMLR*, 2003.
[11] Lei Tang, Jianhui Chen, and Jieping Ye. On multiple kernel learning with multiple labels. IJCAI, 2009.
[12] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis P. Vlahavas. Random k-labelsets for multilabel classification. *IEEE Trans. Knowl. Data Eng.*, 2011.
[13] Sihong Xie, Xiangnan Kong, Jing Gao, Wei Fan, and Yu Philip.S. Multilabel consensus classification. http://arxiv.org/abs/1310.4252.
[14] Min-Ling Zhang and Kun Zhang. Multi-label learning by exploiting label dependency. KDD, 2010.
[15] Xiatian Zhang, Quan Yuan, Shiwan Zhao, Wei Fan, Wentao Zheng, and Zhong Wang. Multi-label Classification without the Multi-label cost. In *SDM*, 2010.