Multi-label Collective Classification in Multi-attribute Multi-relational Network Data

Priyesh Vijayan
Ericsson Research
Chennai, India
Email: priyesh@hotmail.co.in

Shivashankar Subramanian
Ericsson Research
Chennai, India
Email: s.shivashankar@ericsson.com

Balaraman Ravindran
Department of CSE, IIT Madras
Chennai, India
Email: ravi@cse.iitm.ac.in

***Abstract*—Classical machine learning techniques assume the data to be i.i.d., but the real world data is inherently relational and can generally be represented using graphs or some variants of a graph representation. The importance of modeling relational data is evident from its increasing presence in many domains: Telecom networks, WWW, social networks, organizational networks, images, protein sequences, etc. This field has recently been receiving a lot of attention in various communities under different themes depending on the problem addressed and the nature of solution proposed. Collective classification is one such popular approach which involves the use of a local classifier that embeds the node's own attributes and neighbors' information in a feature vector, and classifies the nodes in an iterative procedure. Despite the increasing popularity, there is not much attention paid towards datasets with multiple attributes and multi-relational (MAMR) networks under multi-label scenarios. In MAMR data, nodes can be represented using multiple types of attributes (attribute views) and there are multiple link types between the nodes. For example, in Twitter, users can be represented using their tweets, urls shared, hashtags and list memberships. And different Twitter users can be connected using follower, followed by and re-tweet links. Secondly, in many networks, nodes are associated with more than one label. For instance, Twitter users can be tagged with one or more labels from a set L, where L contains various movie genres that a user might like. Motivated by this, we propose a learning technique for multi-label collective classification using multiple attribute views on multi-relational network data which captures complex label correlations within and across attribute/relationship types. We empirically evaluate our proposed approach on Twitter and MovieLens datasets, and we show that it performs better than the state-of-art approaches.**

I. Introduction

Many real world applications such as web page classification [1], churn prediction [2] and sentiment analysis [3] etc., have an inherent network structure that results in correlation between labels of neighboring data points. For example, in web page classification, hyperlinks between web pages convey that there is a strong correlation between labels of linked pages. Existing works that handle network data can be broadly classified into two types: (1) methods that use only relational (structure) information (2) methods that use both attribute and relational information. It has been shown that latter type of methods capture richer information than the former ones.

Collective classification is one of the popular approaches that can handle both attribute and relational information [4], [5]. It includes node classification techniques, which jointly model attribute data and label correlation information of related objects, by combining traditional machine learning and link (neighborhood structure) based classification in an iterative procedure. Semi-supervised collective classification techniques have been proposed by researchers to handle partially labeled networks [3], [6], [7].

All these collective classification techniques assume the data-points to have only one attribute and one link representation. But many real-world datasets possess additional information that can be utilized to improve the performance. For example, in an academic dataset, in order to classify research interests, researchers can be represented not only using a single attribute view such as their publications' text but also using other attribute views such as their homepage content, conference details of their publications and multiple relational views such as their co-authorship network, co-citation network and so on. Thus network data can have multiple attribute (vector-based) and multiple relational (graph-based) representations. Also, in many such complex network datasets such as Twitter, Facebook and LinkedIn, the nodes are commonly associated with more than one label. For instance, the labels could be social network users' interests. This is referred to as multi-label classification [8].

When multiple views of data are available, we can use a different class of methods to take advantage of unlabeled data [9]. Multi-view learning techniques learn a model for each available view of the data and minimize the disagreement between multiple views on the unlabeled data. Co-training [10], [11] is a multi-view semi-supervised learning algorithm which learns a model on each view of the data and exchanges confident predictions with each other, thereby leveraging complementary information available across different views. However, multi-view learning methods do not model network data with relational features.

Recently, $MAMR$ setup and multi-label classification on network data has drawn the attention of researchers separately. To the best of our knowledge there is no existing work for multi-label classification handling $MAMR$ data in a semi-supervised setup. This scenario can be found in many applications such as: (1) Twitter users' interest classification, where users can be represented using their tweets, urls shared, hashtags and list memberships. And different Twitter users can be connected using follower, followed by and re-tweet links. (2) Topic classification on web page dataset, where each web page can be represented using the text, hashtags and images. Additionally different web pages can be connected using in-link, out-link and co-citation networks.

The key challenge would be to design a multi-label semi-supervised learning technique that would not only exploit multiple attribute and relational view data but also correlation between labels. In multi-label scenario, there may be dependency among labels associated with instances as in the case of research interest prediction, where a researcher working on data mining is more likely to work on machine learning, whereas a researcher working on microprocessors is unlikely to work on machine learning and there may be correlation among labels of related instances as researchers with similar interest collaborate together for publications/projects. Thus complex label correlations must be captured within the same instance across multiple attribute views and among labels of related instances across multiple relational views. In this work, we address these challenges by building upon multi-view learning technique to solve multi-label collective classification on $MAMR$ data by treating multiple attribute and multiple relational types as different views.

In this paper, we refer to network data with multiple attribute views (representations) and multi-relational information as multi-attribute multi-relational ($MAMR$) datasets. Also we use 'single attribute' to denote datasets with only one attribute view and 'multi-attribute' for datasets with multiple types of attribute views. Finally, as commonly used in the literature, we also use single and multi-relational data for networks with single and multiple relationship types respectively.

II. Related Works

Some of the recent works which address a similar classification problem are discussed below:

Multi-Label Collective Classification ***(MLCC)*** [12] adapts collective classification technique to handle multi-label classification on single attribute single relational network data. It transforms the multi-label problem into multiple binary relevance problems one for each label and captures complex label correlations that may exist among labels within the same instance and across related instances; by stacking labels of the same instance and related instances with the feature set.

Across-model Collective Ensemble Classification ***(CEC)*** [13] is a single-label collective classification technique for single attribute multi-relational network data. In [13], the authors propose an ensemble framework that can iteratively infer from multiple collective classifiers learnt over multiple networks, one for each network.

Iterative annotation of multi-relational social networks ***(IMR)*** [14] is a multi-label collective classification technique for single attribute multi-relational network data. It treats the multi-label problem as multiple binary relevance problems by learning collective classifiers for each label on the feature set stacked with aggregate label information of related instances from multiple relations for respective label classifiers. This technique does not capture label correlations as [12].

Heterogeneous Learning ***(GBDT)*** [15] is a single label classification technique for multi-attribute multi-relational network data. It is an error driven model which constructs a function on each attribute view and tries to globally reduce an empirical error function with two constraints: (1) Consensus across various attribute sources (2) Connected instances should have similar predictions.

[12], [13] and [14] are iterative inference techniques (transductive setup) while [15] is a semi-supervised inductive learning technique. Our focus is on inductive learning for multi-label classification on $MAMR$ data that should capture complex label correlations. The closest related work would be [15], but since it enforces homophily and does not leverage label correlation information, it cannot be directly used for multi-label collective classification.

III. Problem Definition

In this section we define the problem of multi-label classification on $MAMR$ data and list down key challenges involved in addressing the problem. Conventional node classification algorithms (for single-label classification) in partially labeled networks propagate labels among nodes until convergence. In this setup, label information for a subset of nodes will be completely known and that of the remaining nodes would be unknown. But in many social network datasets such as Facebook, users may be associated with multiple groups (multi-label classification). In this scenario, label information will not be completely known even for a subset of the nodes, i.e., not all label assignments will be known for nodes. For each group (label), we could generate a labeled set of nodes, based on any of the labeling strategies, such as group membership based label assignment. Thus we learn a classifier for each label separately that models the complex data and also captures label correlations effectively.

The dataset is represented as $D(N, A, G, L, T, U, Y)$, where N is the number of instances(nodes), A is the set of vector based attribute views, G is the set of graph based relational features, L is the label set, T is the family of sets of labeled instances' indices for each label, U is the family of sets of unlabeled instances' indices for each label and Y is the label vector for instances. Important notations followed in this paper are tabulated in Table I.

TABLE I Symbol Table

Symbol	Definition
$A = \{A^1, A^2, \ldots, A^p\}$	set of p attribute views ($A_i \in \mathbb{R}^{j_i}$, j_i is the dimension of i^{th} view)
$G = \{G^1, G^2, \ldots, G^q\}$	set of q relational(graph) views
$L = \{L_1, ...L_k\}$	the set of k labels
$T = \{T_1, T_2, \ldots, T_k\}$	k sets of labeled instances' indices over N
$U = \{U_1, U_2, \ldots, U_k\}$	k sets of unlabeled instances' indices over N
$Y_i = (Y_i^1, Y_i^2, \ldots, Y_i^k)$	label vector for i^{th} instance, ($Y_i^j = 1$ if $j \in L$ else $Y_i^j = 0$)

In order to exploit $MAMR$ data for multi-label classification, various information need to be captured while modeling. The key challenges involved are:

1) Building a unified model for multiple views which may differ in representation (attribute and structure) and statistical properties (distribution), $P(Y|A, G)$.
2) Maximizing consensus among multiple attribute and multiple relational views separately to leverage the unlabeled information. Also, exploiting complementary information between attribute and link views at a higher level that effectively utilizes both the node's profile and relationships.
3) Capturing complex label correlations that may exist within the same instance across multiple attribute views and among labels of related instances across multiple relational views, also (at a higher level) between attribute and relational views.

$$min(\underbrace{L_1(A,G,L,T)}_{\text{Loss on labeled data}}+\underbrace{L_2(A,G,L,U)+L_3(A,G,L,U)}_{\text{Loss on unlabeled data}}) \tag{1}$$

$$L_1(A,G,L,T)=\sum_{l\in L}\Big(\underbrace{\sum_{a\in A}\mathfrak{L}(f_a^l(A_{T_l}^a),Y_{T_l}^l)}_{\text{Attribute views' disagreement}}+\underbrace{\sum_{g\in G}\mathfrak{L}(f_g^l(G_{T_l}^g),Y_{T_l}^l)}_{\text{Relational views' disagreement}}\Big)$$

$$L_2(A,G,L,U)=\sum_{l\in L}\Big(\underbrace{\sum_{a\in A}(f_a^l(A_{U_l}^a)-\prod_{a\in A}f_a^l(A_{U_l}^a))^2}_{\text{Disagreement among attribute views}}+\underbrace{\sum_{g\in G}f_g^l(G_{U_l}^g)-\prod_{g\in G}f_g^l(G_{U_l}^g))^2}_{\text{Disagreement among relational views}}\Big)$$

$$L_3(A,G,L,U)=\sum_{l\in L}\Big(\underbrace{\sum_{a\in A}(f_a^l(A_{U_l}^a)-\prod_{g\in G}f_g^l(G_{U_l}^g))^2}_{\text{Attribute views' disagreement with relational views}}+\underbrace{\sum_{g\in G}f_g^l(G_{U_l}^g)-\prod_{a\in A}f_a^l(A_{U_l}^a))^2}_{\text{Relational views' disagreement with attribute views}}\Big)$$

IV. Proposed Solution

A. Proposed Framework

Semi-supervised learning methods minimize the loss on both labeled and unlabeled data. The loss on labeled data is solved with supervised machine learning techniques that fit a model to the data, whereas the loss on unlabeled data varies depending on the semi-supervised paradigm used. In the case of disagreement based semi-supervised techniques such as co-training methods, loss on unlabeled data is captured as the disagreement on unlabeled data between views.

We propose a co-training style learning framework for multi-label classification on multi-attribute and multi-relational data. The proposed framework learns a classifier (f^l) for each label, ($l \in L$) on each attribute view ($a \in A$) and each relational view ($g \in G$), leveraging both labeled and unlabeled data. The classifiers learned on attribute and relational views are denoted as f_a^l and f_g^l respectively. The objective function solved by the framework can be expressed as a minimization of three loss terms L_1, L_2 and L_3 as given in equation (1).

L_1 represents the loss on labeled data on both attribute and relational views, where L is the loss function of the classifier used. For SVM classifier [16] which we use in this work, $\mathfrak{L}$ would be Hinge loss. The proposed framework leverages unlabeled data by reducing the loss terms L_2 and L_3, where L_2 represents the local disagreement within predictions of multiple attribute and multiple relational views and L_3 represents the global disagreement between the predictions of individual attribute and relational views with combined predictions from relational and attribute views respectively.

The proposed co-training style framework handles L_2 and L_3 losses with two update steps alternatively. The first update uses an ensemble averaging based co-training method to maximize consensus locally within multiple attribute views and multiple relational views. In ensemble averaging based co-training, we learn a model on each view and combine the predictions using voting. We select instances with high scores from voting to be appended to the labeled set for re-training. Conventional co-training algorithms re-train from confident predictions of individual views which may be noisy, whereas ensemble based techniques can help in reducing variance (error on unlabeled data) which is very critical for semi-supervised setup in order to prevent noisy label propagation [17]. The second update uses a co-training style method to reduce the disagreement between attribute and relational views globally by exchanging highly confident predictions obtained from ensemble of local view classifiers from the previous update.

B. Handling Relational Information

Co-training methods cannot handle relational data directly; hence we need to transform relational views into vector-based views. Each relational view, G^i is transformed into a vector space, (TG^i) where instances are represented with aggregated label information [5] (COUNT, MODE, label distribution, etc.) of it's neighbors in G^i. This transformation enables us to learn a unified model across multiple views (attribute and relational). Secondly, co-training assumes that the instances do not have any missing views, but $MAMR$ data could have missing views especially relational views since it can contain arbitrary relationships between nodes. Nodes with arbitrary relationships, i.e., not all relationship types might be present for a particular node, can easily be found in social networks. For instance, in case of Twitter, we witness absence of follower information for less active users. In such cases, it is not advisable to consider missing views for nodes while learning. The transformation step handles such cases suitably for different aggregation techniques used: zero vector for count based transformation, equal probability vector for label distribution based transformation, and so on.

C. Handling Label Correlation

$MAMR$ data has complex label correlations. Labels within the same instance may be correlated across different attribute views and labels of related instances may be correlated across different relational views. In order to capture label correlation information for each label within instances, we follow a two step procedure similar to [18]. In the first step, for each label we predict labels for instances on each view and combine them by voting. In the second step, for each label l on each view A^i, we obtain a new feature set, SA^i by stacking A^i with a binary vector Y_i^{-l} that captures the labels in $\{L-l\}$ obtained from the predictions. Since the same label correlation information Y_i^{-l} for each label l obtained from voting is stacked across multiple attribute views, consensus among stacked attribute views will be maximized. Whereas label correlations between related instances are automatically captured by classifiers learned on the transformed relational data, as the data explicitly represents the summarized label correlation information of related instances.

D. Algorithm

The objective function in equation (1) is extended to capture relational information and label correlations as mentioned in section IV-B and IV-C respectively. The new objective function is given below in equation (2).

$$min(L_1(SA, TG, L, T) + L_2(SA, TG, L, U) + L_3(SA, TG, L, U)) \quad (2)$$

The proposed solution handles the above mentioned objective function as described in Algorithm 1. In order to learn on multiple views with label correlation information, we need label predictions for stacking as explained in IV-C. Hence we bootstrap labels for unlabeled instances using multi-view learning on attribute views as explained by the function MVLearning in the Algorithm 2. After bootstrapping, the iterative multi-view learning procedure follows. We minimize the four disagreement terms expressed in L_2 and L_3 losses alternatively and iteratively by the following four steps:

1) We create stacked attribute views for each labels as explained in section IV-C ($stackLabelCorrelation(A, Y, l)$). For unlabeled instances, stacked attribute views are created by stacking the original feature set of views with label predictions obtained from an ensemble (voting) of attribute learners and for labeled instances given labels are stacked. Then for each label, we (re-)train a model on each attribute view with top confident predictions from an ensemble of attribute classifiers (either from step 4 or initially with bootstrapped ensemble of predictions). This step aims to locally maximize consensus within multiple attribute views.
2) We create relational views as explained in section IV-B ($transformGraph(G, Y)$) with predictions from an ensemble of attribute classifiers. Then for each label, we (re-)train a model on each transformed relational view with top confident labels shared by an ensemble of attribute classifiers from step 1. This step aims to globally reduce the disagreement between each relational view and multiple attribute views by learning from shared complementary information from an ensemble of attribute classifiers.
3) We update the transformed relational views ($transformGraph(G_T, Y_T)$) with confident predictions from step 2. For each label, we (re-)train a model on each relational view with top confident predictions from an ensemble of relational classifiers from step 2. This step aims to locally maximize the consensus within multiple relational views.
4) We update stacked attribute views ($stackLabelCorrelation(A_T, Y_T, l)$) with confident predictions from step 3. For each label, we (re-)train a model on each stacked attribute view with top confident predictions shared from an ensemble of relational classifiers and thereby reduce the global disagreement between each attribute view and multiple relational views. Steps 1 to 4 are iterated until the termination condition is met.

MVLEARNING$(views, T, U, Y)$ function learns a one-vs-all SVM classifier for each label on the labeled data obtained from an ensemble of multi-view learners either locally within attribute/relational views or globally across attribute and relational views. The posterior probability for positive class and negative class for each label on unlabeled data is predicted on all views. Then we combine probabilities by voting to find highly confident predictions for each label and append it to the labeled set for that label.

Algorithm 1 $ML - MAMR$

```
Input: D(A, G, L, T, N, Y)
Output: [f_A, f_G, Y]
Bootstrap:
  for l in L do
    [f_A, T^l, U^l, Y^l] = MVLearning(A, T^l, U^l, Y^l)
  end for
  Iterative multi-view learning:
  repeat
  Step 1:
    for l in L do
      SA = stackLabelCorrelation(A, Y, l)
      [f_A^l, T^l, U^l, Y^l] = MVLearning(SA, T^l, U^l, Y^l)
    end for
  Step 2:
    TG = transformGraph(G, Y)
    for l in L do
      [f_G^l, T^l, U^l, Y^l] = MVLearning(TG, T^l, U^l, Y^l)
    end for
  Step 3:
    TG_T = transformGraph(G_T, Y_T)
    for l in L do
      [f_G^l, T^l, U^l, Y^l] = MVLearning(TG, T^l, U^l, Y^l)
    end for
  Step 4:
    for l in L do
      SA_T = stackLabelCorrelation(A_T, Y_T, l)
      [f_A^l, T^l, U^l, Y^l] = MVLearning(SA, T^l, U^l, Y^l)
    end for
  until U^l = ϕ, ∀l ∈ L
```

Algorithm 2 MVLearning

```
function MVLEARNING(views, T, U, Y)
  for X in views do
    f_X = SVM_Train(X_T, Y_T)
    [P_X, N_X, Y_u] = SVM_Predict(f_X, X_U)
  end for
  P_Prob = Π_X P_X, where x ∈ A [voting]
  N_Prob = Π_X N_X, where x ∈ A [voting]
  T = T ∪ getConfidentLabels(P_Prob, N_Prob)
  U = U \ T
return [f, T, U, Y]
end function
```

The $getConfidenceLabels(P_{Prob}, N_{Prob})$ function returns instances with top confident predictions. There are many techniques to choose confident labels. We choose the top 10% confident predictions while maintaining the label distribution (positive, negative class distribution for each label) same as in the given labeled set.

For collective inference on $MAMR$ data under semi-supervised conditions, we can use the same iterative learning procedure given in Algorithm 1 without re-estimating the parameters of f_a^l and f_g^l.

V. Experimental Results

A. Datasets

We have used two datasets to evaluate the performance of the proposed $ML-MAMR$ model. A short description of the datasets used is given below:

Rugby Players and Clubs on Twitter (Twitter Dataset):
UCD MLG group's multi-view Twitter dataset[1] is a collection of *854 International Rugby Union players*, clubs and organizations on Twitter. The ground truth consists of communities corresponding to *15 countries*. The communities are overlapping, as players can be associated with their home nation and the nation in which they play club rugby. We used all the views as it is from the source dataset.
There are 9 different views in the dataset, viz.:
Attribute views (3): tweet contents of players, list memberships of each player and the corresponding lists' contents.
Relational views (6): followers, followedby, mentions, mentionedby, retweets and retweetedby relations of players.
The characteristics of the data are as follows:
Instances: 854; ***Labels:*** 15; ***Label cardinality:*** 1.2307
Label density: 2.2976

MovieLens Dataset (Movie Dataset):
This dataset is an extension of GroupLens research group's MovieLens10M[2] dataset. The task here is to predict relevant genres for each movie.
We extracted 4 different views from the data, viz.:
Attribute views (2): movie summary and movie tags.
Relational views (2): actor and director graphs.
Tags of movies are directly obtained from the source dataset, while summary of movies were extracted from IMDB[3] database using the IMDB ids of movies given in the source dataset. Summary (text) is represented using term-frequency (TF) representation, where the vocabulary was built using distinct words with (freq(words)≻3). Actor and director information of movies are available in the source dataset, with which we created actor and director graphs by adding links between movies that share common artists and directors respectively.
The characteristics of the data are as follows:
Instances: 3911; ***Labels:*** 18; ***Label cardinality:*** 0.0820;
Label density: 0.1276

In general, the number of labels associated to instances (on an average) plays a key role in capturing label correlations. We can see that Movie dataset has higher label density and cardinality measures compared to Twitter dataset which allows the proposed technique to capture richer correlation information. This can also be seen in experimental results section, where the gain in performance using the proposed method (compared to state-of-art techniques) is higher on Movie-Genre dataset.

B. Baseline methods and Experimental Setup

We compare our proposed $ML-MAMR$ learning approach with three related works discussed in section II, viz: MLCC [12], CEC [13] and GBDT [15]). Our primary focus is on capturing complex label correlations (within instance and across related instances) besides $MAMR$ setup and since IMR [14] assumes all labels to be independent, we use CEC_ML as a baseline to compare our performance with a technique that handles single-attribute multi-relational data with complex label correlations. In CEC_ML, we stack label correlation information similar to MLCC in CEC method.

In order for $MLCC$, CEC and CEC_ML to handle multi-attribute data we combine multiple attribute views into a single attribute view by stacking views together and similarly for MLCC to handle multi-relational data we combine multiple relational views into a single relational view by adding links between nodes if they have at least one link (based on any of the relationships) between them.

$ML-MAMR$, $MLCC$, CEC and CEC_ML do not have any parameters. As our work does not focus on weighting views, in GBDT we give equal weight to attribute views and relational views on an abstract level ($\lambda_0 = 0.5$ and $\lambda_1 = 0.5$) and also at individual view level among attribute and relational views (w).The base classifier used for the experiments is libsvm's [16] implementation of SVM.

The below mentioned performance measures are averaged using 5-fold cross validation for each labeled ratio. We repeat the experiments with different labeled ratios (10% 30% 50% 70% and 90%) in-order to evaluate the robustness of our proposed method under label sparsity conditions. For each label, training instances are chosen by using stratified random sampling, i.e., instance ratio for each label in the labeled set is maintained at the same ratio as present in the entire dataset [19]. We evaluate the performance using four commonly used metrics for multi-label classification [8]: Exact-Match-Ratio, Accuracy, Precision and Hamming Loss.

C. Results

Experimental results on the two datasets comparing our proposed approach with baseline methods are given in Tables II and III. The results are given on a percentage scale. Performances of the best method on each metric are highlighted. Overall, the proposed approach performs better than the baseline approaches. Some of the observations that we derived from the results are given below:

- Performance gain on Movie dataset is higher compared to Twitter dataset, which goes well with the intuition that it is possible to capture label correlations better on a dataset which has higher label density/cardinality.
- Following from the previous observation, on Twitter dataset, $GDBT$ performs better than other baseline techniques and similarly on Movie dataset, $MLCC$ performs better than the rest. It reinstates the argument that $GDBT$ assumes homophily and is not naturally suitable for multi-label classification.
- Proposed approach shows better gain in performance for scenarios with lower labeled instances (training ratio), i.e., handles label sparsity by leveraging all the views effectively.
- Exact match ratio is considered as the strictest evaluation metric for multi-label classification. From the experimental results, we can see that the proposed approach shows maximum gain in performance on exact match ratio than other metrics. It re-emphasizes the fact that our proposed approach captures necessary label correlation information effectively.

[1] http://mlg.ucd.ie/networks/rugby.html, http://mlg.ucd.ie/aggregation/
[2] http://ir.ii.uam.es/hetrec2011/datasets.html
[3] http://www.imdb.com

TABLE II TWITTER DATASET (RUGBY-NATION)

	10%	30%	50%	70%	90%
Exact Match Ratio					
MLCC	0	0	27.811	79.291	85.872
CEC	49.161	70.1	77.339	78.085	78.234
CEC_ML	47.364	72.743	80.344	83.42	84.985
GBDT	56.547	70.704	77.58	82.416	83.944
ML-MAMR	**65.319**	**82.598**	**85.029**	**87.791**	**88.276**
Accuracy					
MLCC	15.238	24.382	57.868	87.406	90.7
CEC	66.636	79.93	83.246	83.783	83.503
CEC_ML	66.991	82.642	86.954	88.727	89.168
GBDT	66.353	79.688	84.681	87.721	88.871
ML-MAMR	**76.821**	**89.579**	**91.055**	**92.487**	**92.038**
Precision					
MLCC	15.447	24.663	58.991	89.268	91.929
CEC	70.074	82.633	85.536	85.726	84.913
CEC_ML	70.593	85.489	89.58	90.733	90.397
GBDT	70.998	83.758	87.652	89.936	90.143
ML-MAMR	**82.962**	**94.612**	**94.889**	**95.186**	**92.976**
Hamming Loss					
MLCC	46.083	23.575	7.632	1.542	1.044
CEC	5.449	2.7	1.861	1.707	1.641
CEC_ML	5.099	2.301	1.532	1.236	1.116
GBDT	3.587	2.306	1.716	1.313	1.129
ML-MAMR	**3.023**	**1.42**	**1.195**	**0.955**	**0.898**

TABLE III MOVIE DATASET (MOVIE-GENRE)

	10%	30%	50%	70%	90%
Exact Match Ratio					
MLCC	0.997	1.07	1.637	17.85	22.523
CEC	7.189	10.825	12.045	11.746	12.712
CEC_ML	4.544	4.898	7.479	12.9	13.921
GBDT	3.37	3.336	3.951	4.974	6.476
ML-MAMR	**14.849**	**17.565**	**20.306**	**22.413**	**24.683**
Accuracy					
MLCC	12.163	12.556	13.556	37.222	38.609
CEC	29.924	32.649	32.247	31.276	29.255
CEC_ML	23.326	24.419	27.978	34.449	36.178
GBDT	30.45	32.146	33.35	33.533	32.447
ML-MAMR	**32.755**	**36.635**	**39.216**	**39.672**	**39.232**
Precision					
MLCC	14.243	15.671	17.06	53.323	49.268
CEC	42.889	47.502	46.257	43.12	36.681
CEC_ML	32.028	33.827	38.704	46.353	43.899
GBDT	37.335	38.017	38.988	38.738	36.355
ML-MAMR	**55.003**	**58.137**	**59.738**	**57.233**	**50.435**
Hamming Loss					
MLCC	48.632	35.64	34.708	10.588	9.33
CEC	15.985	13.559	12.993	12.454	11.642
CEC_ML	24.48	18.494	15.339	12.195	11.02
GBDT	20.249	19.535	18.222	17.285	16.303
ML-MAMR	**11.815**	**10.981**	**10.304**	**9.817**	**9.116**

VI. CONCLUSION

In this paper we studied the problem of multi-label classification for multi-attribute multi-relational data sources. The complexities of the data require a unified multi-label model to learn from multiple attribute and arbitrary relational views. Secondly, the proposed technique also captures various complex label correlations within and across attribute and relational data. To the best of our knowledge, the proposed co-training style algorithm is the first to provide a solution to this problem. The proposed algorithm tries to maximize the consensus among various attribute and relational views individually, and simultaneously reduces disagreement between attribute and relational views by sharing complementary information between them. It is very much evident from the empirical results that the proposed algorithm not only exploits multiple views, but also captures label correlations effectively than any of the existing works.

REFERENCES

[1] L. K. McDowell et al, "Cautious collective classification," in *The Journal of Machine Learning Research,* vol.10, pp. 2777–2836, 2009.

[2] R. J. Oentaryo et al, "Collective churn prediction in social network," in *Proc. 2012 International Conference on Advances in Social Networks Analysis and Mining. IEEE Computer Society,* 2012, pp. 210–214.

[3] S. Shivashankar and B. Ravindran, "Multi Grain Sentiment Analysis using Collective Classification," in *Proc. 19th European Conference on Artificial Intelligence,* 2010, pp. 823–828.

[4] P. Sen et al, "Collective Classification in Network Data," in *AI Magazine,* vol.29, no.3, pp.93–106, 2008.

[5] J. Neville and D. Jensen, "Iterative Classification in Relational Data," in *Proc. Association for the Advancement of Artificial Intelligence: Workshop on Learning Statistical Models from Relational Data,* 2000, pp.13–20.

[6] Q. Lu and L. Getoor, "Link-based Classification using Labeled and Unlabeled Data," in *Proc. International Conference on Machine Learning: Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining,* 2003, pp.496–503.

[7] X. Zhu, "Semi-supervised learning literature survey," in *Tech. Report 1530, Department of Computer Sciences, University of Wisconsin, Madison,* 2005

[8] G. Tsoumakas et al, "Mining multi-label data," in *Data mining and knowledge discovery handbook, Springer US,* pp. 667–685, 2010.

[9] C. Xu et al "A survey on multi-view learning," in *ArXiv preprint,* vol.1304, no.5634, 2013.

[10] A. Blum and T. M. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th annual conference on Computational Learning Theory,* 1998, pp.92–100.

[11] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *Proc. 9th international Conference on Information and Knowledge Management,* 2000, pp. 86–93.

[12] X. Kong et al, "Multi-Label Collective Classification," in *SIAM International Conference on Data Mining(SDM),* 2011, pp. 618–629.

[13] H. Eldardiry and J. Neville, "Across-model collective ensemble classification," in *Proc. 25th Association for the Advancement of Artificial Intelligence conference on Artificial Intelligence,* 2011.

[14] S. Peters et al, "Iterative annotation of multi-relational social networks," in *Proc. 2010 International Conference on Advances in Social Networks Analysis and Mining. IEEE Computer Society,* 2010, pp. 96–103.

[15] X. Shi et al, "Learning from Heterogeneous Sources via Gradient Boosting Consensus," in *SIAM International Conference on Data Mining(SDM),* 2012, pp. 224–235.

[16] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," in *ACM Transactions on Intelligent Systems and Technology,* vol.2, no.3, pp. 27.1–27.27, 2011.

[17] J. Tanha et al, "Disagreement-based co-training," in *Proc. 23rd IEEE International Conference on Tools with Artificial Intelligence,* 2011, pp. 803–810.

[18] S. Godbole, and S. Sarawagi, "Discriminative Methods for Multi-labeled Classification," in *Advances in Knowledge Discovery and Data Mining, Springer Berlin Heidelberg,* pp. 22–30, 2004.

[19] K. Sechidas, "On the stratification of multi-label data," in *Machine Learning and Knowledge Discovery in Databases: Springer Berlin Heidelberg,* pp. 145–158, 2011.

[20] D. Greene and P. Cunningham, "Producing a Unified Graph Representation from Multiple Social Network Views," in *5th Annual ACM Web Science Conference,* 2013, pp. 118–121.