Multi-Label Relational Neighbor Classification using Social Context Features

Xi Wang
Department of EECS
University of Central Florida
Orlando, Florida, USA
xiwang@eecs.ucf.edu

Gita Sukthankar
Department of EECS
University of Central Florida
Orlando, Florida, USA
gitars@eecs.ucf.edu

ABSTRACT

Networked data, extracted from social media, web pages, and bibliographic databases, can contain entities of multiple classes, interconnected through different types of links. In this paper, we focus on the problem of performing multi-label classification on networked data, where the instances in the network can be assigned multiple labels. In contrast to traditional content-only classification methods, relational learning succeeds in improving classification performance by leveraging the correlation of the labels between linked instances. However, instances in a network can be linked for various causal reasons, hence treating all links in a homogeneous way can limit the performance of relational classifiers.

In this paper, we propose a multi-label iterative relational neighbor classifier that employs social context features (SCRN). Our classifier incorporates a class propagation probability distribution obtained from instances' social features, which are in turn extracted from the network topology. This class-propagation probability captures the node's intrinsic likelihood of belonging to each class, and serves as a prior weight for each class when aggregating the neighbors' class labels in the collective inference procedure. Experiments on several real-world datasets demonstrate that our proposed classifier boosts classification performance over common benchmarks on networked multi-label data.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database applications—*Data Mining*; J.4 [**Social and Behavior Sciences**]: Sociology

General Terms

Algorithm, Experimentation

Keywords

Relational learning; Collective classification; Social Dimensions

1. INTRODUCTION

Recently, much attention has been paid to the problem of learning from networked data, where instances are interconnected by implicit or explicit relationships [1, 5, 21]. Relational learning [6] can learn models of this data structure by utilizing the correlation between labels of linked objects; networks resulting from social processes often possess a high amount of homophily, such that nodes with similar labels are more likely to be connected [15]. Many of the algorithms developed for relational classification are heuristic methods that do not necessarily correspond to formal probabilistic semantics [17]. In other approaches, during the inference process the probability distribution is structured as a graphical model based on the assumption that the structure of the network corresponds at least partially to the structure of the network of probabilistic dependencies [14]. Relational learning enhances the tractability of estimating the full joint probability distribution of the data by making a first-order Markov assumption that the label of one node is dependent on that of its immediate neighbors in the graph. Collective inference in relational classification makes simultaneous statistical estimations of the unknown labels for interrelated entities, and finds a equilibrium state such that the inconsistency between neighboring nodes is minimized. By exploiting network connectivity information, relational classification models have been shown to outperform traditional classifiers [18, 25].

The conventional relational classification model focuses on the single-label classification problem, which assumes that each instance is only associated with one label among a finite set of candidate classes. However, in many real relational datasets, each instance is associated with multiple labels. For instance, in document networks, one document can describe multiple topics. In social networks, people often belong to a large set of interest groups. Classifying this type of dataset can be regarded as a multi-label classification task. In previous work, edges in the network are treated homogeneously; the implicit assumption is that the edges are engendered from similar social processes. However, in multi-label relational datasets, connections between instances are driven by various casual reasons. In the familiar example of collaboration networks, scientific authors usually have multiple research interests and seek to collaborate with different co-authors for different types of work. For instance, Author A cooperates with author B on publishing papers in machine learning conferences whereas his/her interaction with author C is mainly due to work in the data mining area. The heterogeneity in connection causality makes the classification problem more difficult.

Collective classification becomes particularly challenging in multi-label settings since the label dependencies among related instances are more complex. Currently, most collective inference models do not differentiate in their treatment of connections between instances; however, treating links in a homogeneous way may negatively affect the classification result [23]. The relational neighbor classifier (*RN*) [13] provides a simple yet effective way to solve

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.

single-label relational classification problems. In this paper, we present a multi-label relational classifier that accounts for this inhomogeneity in connections and is designed for classification problems on multi-label networked datasets. Our proposed method, *SCRN*, extends *RN* by introducing a node class-propagation probability that modulates the amount of propagation that occurs in a class specific way based on the node's similarity to each class. Although the class propagation probability can be determined by the node's intrinsic features, it can also be based on node's social features. These features capture link patterns between a node and its neighbors and can be extracted from network topology in instances when the node lacks intrinsic features. *SCRN*'s ability to differentiate between classes during the inference procedure allows it to outperform previous methods in several real-world multi-label relational datasets.

The multi-label collective classification problem that we address here is related to the within-network classification problem: entities whose labels are known are linked to entities for which the class has to be estimated. In this paper, we aim to simultaneously predict the label sets of a group of related instances within the same network. The multi-label networked dataset is represented as a graph $G = (V, E, C, L)$, where $V = \{v_1, v_2, ..., v_n\}$ is a set of nodes, E is a set of edges that connect pairs of nodes. Let $C = \{c_1, c_2, ..., c_m\}$ be the finite set of m possible classes that each node can possess. Given a node $v_i \in V$, its class label is represented by a binary vector $L_i = (l_i^1, l_i^2, ..., l_i^m) \in \{0, 1\}^m$, indicating the multiple label assignment to each node. $l_i^m = 1$ iff v_i belongs to class c_m. The set of nodes, V, is further divided into two disjoint parts: nodes with known class labels, V^K, and V^U, nodes whose labels need to be determined. $L_K = \{L_i | v_i \in V^K\}$ indicates the observed multi-label set assigned to V^K. Our task is to use V^K as the training data to infer the labels, L_U, for nodes in V^U.

In multi-label classification problems, a popular approach is to decompose the multi-label classification problem into multiple binary classification problems (one for each class). Conventional multi-label classification approaches (e.g., the ones used on non-networked data), usually assume the instances are i.i.d., and that the inference for each instance is performed independently:

$$P(L|V) \propto \prod_{v_i \in V^U} P(L_i|v_i). \quad (1)$$

In this paper, we propose a multi-label relational classifier that models the correlations between inter-related instances in the network. We start by constructing a social feature space, an edge-based representation of social dimensions using the network topology to capture the node's potential affiliations as described in [23]. A class-propagation probability is assigned to each node to describe its intrinsic correlation to each class. The class-propagation probability is calculated from the similarity between the node's social features and the class reference vector. The multi-label relational classifier estimates a node's label set based on its neighbors' class labels, the similarity between connected nodes, and its class propagation probability. *SCRN* iteratively classifies the labels of the unlabeled nodes until all the label predictions are fixed or the maximum number of iterations is reached. In the next section, we describe the basic idea behind relational neighbor classifiers before describing our proposed method.

2. APPROACH

The Relational Neighbor (*RN*) classifier proposed by [13], is a simple relational probabilistic model that makes predictions for a given node based solely on the class labels of its neighbors, without machine learning or additional features. RN estimates class-membership probabilities by assuming the existence of homophily in the dataset, entities connected to each other are similar and likely belong to the same class. Suppose each instance in the network only belongs to a single class $c \in C$. Given $v_i \in V^U$, the relational-neighbor classifier estimates $P(L_i = c|v_i)$, the class-membership probability of a node v_i belonging to class c, as the (weighted) proportion of nodes in the neighborhood that belong to the same class. We define neighbors N_i as the set of *labeled* nodes that are linked to v_i. Thus:

$$P(L_i = c|v_i) = \frac{1}{Z} \sum_{v_j \in N_i} w(v_i, v_j) \times I(L_j = c), \quad (2)$$

where $Z = \sum_{v_j \in N_i} w(v_i, v_j)$. $w(v_i, v_j)$ is the weight of the link between node v_i and v_j and $I(.)$ is an indicator variable.

Instead of making a hard labeling during the inference procedure, the weighted-vote relational neighbor classifier (*wvRN)* extends *RN* by tracking changes in the class membership probabilities. *wvRN* estimates $P(L_i|v_i)$ as the (weighted) mean of the class membership probabilities of the entities in the neighborhood (N_i):

$$P(L_i = c|v_i) = \frac{1}{Z} \sum_{v_j \in N_i} w(v_i, v_j) \times P(L_j = c|N_j), \quad (3)$$

where Z is the usual normalization factor.

In both *RN* and *wvRN*, entities whose class labels are unknown are either ignored or are assigned a prior probability, depending on the choice of the local classifier. Since only a small portion of the nodes in G have known labels, a collective inference procedure is needed to propagate the label information through the network to related instances, using either the *RN* classifier or *wvRN* classifier in its inner loops.

As shown in [13], both *RN* and *wvRN* perform surprisingly well on relational datasets, even compared to more complex models, such as the Probabilistic Relational Model and Relational Probabilistic Tree.

2.1 Proposed Method: SCRN

wvRN assumes that each node only has one single label, and that the class labels of linked nodes are likely to be the same. However, in multi-label relational networks, the existence of heterogeneous relationships gives rise to nodes with neighbors from multiple classes. The diversity of the connections indicates two connected nodes might only share a subset of labels. The inference procedure in the *RN* classifier and *wvRN* classifier treat all links homogeneously, and this may cause problems when propagating the label information across the network, especially when collective inference originates from the overlapping nodes (nodes with multiple labels) in the network. A toy example with two class labels is shown in Figure 1. Imagine the case where all the nodes on the left-hand side of node 1 belong to group 1, while those on the right-hand side of node 1 are from group 2; node 1 serves as a bridge, weakly connecting both groups. If we commence inferring the label sets of all the other nodes using node 1's label information, without differentiating between the connections, the collective inference in *RN* classifier will expect all the nodes in the graph to have the same class label as node 1.

To address this problem, instead of uniformly aggregating the neighbor's labels along each class like *wvRN* does, we propose to assign each node a class propagation probability distribution, which represents its likelihood of maintaining the neighbor's class label set. A node will be more likely to share a class with neighbors that have a high class-propagation probability. Take the toy graph

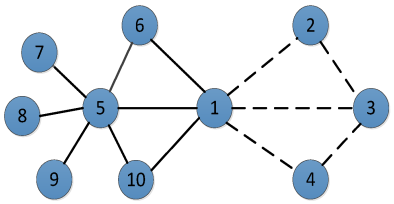

Figure 1: A simple example of a coauthorship network. The solid line represents the act of publishing a paper in a *data mining* conference and the dashed line represents the activity of collaborating on a *machine learning* paper. To express the nodes using edge-based social features, each edge is first represented by a feature vector where nodes associated with the edge denote the features. For instance, here the edge "1-3" is represented as [1,0,1,0,0,0,0,0,0,0]. Then, the node's social feature (SF) vector is constructed based on edge cluster IDs. Suppose in this example the edges are partitioned into two edge clusters (represented by the solid lines and dashed lines respectively), then the SFs for node 1 and 3 become [3,3] and [0,3] using the count aggregation operator.

Table 1: Overview of *SCRN* Algorithm

Input: $\{G, V, E, C, L_K\}$, *Max_Iter*
Output: L_U for nodes in V^U

1. Construct the social feature space using scalable K-means edge clustering.
2. Initialize the class reference vectors, RV, for each class based on Equation 4.
3. Calculate the class-propagation probability for each test node using the similarity between the node's social feature and class reference vectors using the *GHI* kernel.
4. Repeat until # iterations>*Max_Iter* or predictions converge to stable values:
   - Estimate the test node's class membership probability according to Equation 6.
   - Update the test node's class membership probability based on the prediction in the last iteration according to Equation 7.
   - Update the class reference vectors according to the labels of the nodes in the current iteration.
   - Re-calculate each node's class-propagation probability using the present class reference vectors.

for example, when inferring the labels of node 2 from node 1, we want to keep its estimation of class 2's probability much higher than class 1 to make a more accurate prediction. Therefore, a node's class-propagation probability can be regarded as its prior probability for each class. Learning the class-propagation probability distribution is critical in order to achieve better discrimination during the inference procedure. Fortunately the structure of the network can be highly informative, and we capture this information through using the network topology to construct *social features*.

In the proposed method, we first extract the social features (SF) from the network topology using the edge clustering method described in Section 2.2. These social features capture the nodes' involvement patterns in different potential affiliations, and the node's class propagation probability can be constructed from the social features in the following way. An initial set of reference features for class c can be defined as the weighted sum of social feature vectors for nodes known to be in class c:

$$RV(c) = \frac{1}{|V_c^K|} \sum_{v_i \in V_c^K} P(l_i^c = 1) \times SF(v_i), \tag{4}$$

where $V_c^K = \{v_i | v_i \in V^K\}$, which represents the nodes whose labels are known as class c.

Then node v_i's class propagation probability for class c conditioned on its social features, $P_{CP}(l_i^c|SF(v_i))$, can be calculated by the normalized vector similarity between $SF(v_i)$ and class c's reference feature vector, $RV(c)$:

$$P_{CP}(l_i^c|SF(v_i)) = \text{sim}(SF(v_i), RV(c)), \tag{5}$$

where $\text{sim}(a, b)$ is any normalized vector similarity function (e.g., cosine or inner product).

Our proposed multi-label relational classifier then estimates the class-membership probability of node v_i belonging to class c:

$$P(l_i^c|N_i, SF(v_i)),$$

based on the class labels of its neighbors, $\{L_j|v_j \in N_i\}$, the weight between v_i and its directed neighbors v_j, $w(v_i, v_j)$, and its conditional class propagation probability, $P_{CP}(l_i^c|SF(v_i))$. The multi-label relational classifier model is defined as follows:

$$P(l_i^c|N_i, SF(v_i)) = \frac{1}{Z} \sum_{v_j \in N_i} P_{CP}(l_i^c|SF(v_i)) \times w(v_i, v_j) \times P(l_j^c|N_j), \tag{6}$$

where Z is the normalization factor. Similar to the *RN* and *wvRN* classifiers, our multi-label relational classifier iteratively classifies the nodes in V^U using the model defined in Equation 6 in its inner loop. Since the label predictions change in each iteration, the class reference feature vector is updated based on the feature vectors of nodes (both training and testing nodes) whose labels belong to class c in the current iteration. In this paper, we adopt the Relaxation Labeling (RL) approach [3, 27] in the collective inference framework. During each iteration, RL updates the prediction probability by taking account of the probability estimates from the previous iteration. The general update procedure for relaxation labeling is shown in Equation 7 [14]:

$$P_i^{(t+1)} = \beta^{(t+1)} \cdot \mathcal{M}_{\mathcal{R}}(v_i^{(t)}) + (1 - \beta^{(t+1)}) \cdot P_i^{(t)}, \tag{7}$$

where $\beta^{(0)} = k$ and $\beta^{(t+1)} = \beta^{(t)} \cdot \alpha$. Both k and α are constants in the range 0 to 1; t is the iteration count and $\mathcal{M}_{\mathcal{R}}(\cdot)$ denotes the relational model. The inference procedure in *SCRN* terminates when it meets the stopping criteria; possible stopping criteria include the stability of all label predictions between iterations or reaching a fixed budget of iterations. A summary of the *SCRN* framework is shown in Table 1.

2.2 Edge-Based Social Feature Extraction

The notion of edge-based social dimensions was created to address the classification problem in networked data with multiple types of links. Connections in human networks are mainly affiliation-driven, and each connection can often be regarded as principally resulting from one affiliation. Hence, links (connections) possess a strong correlation with affiliation classes. Moreover, since each person usually has more than one connection, the involvements of potential groups related to one person's edges can be utilized as a

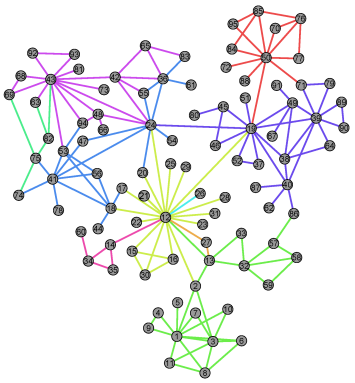

Figure 2: Visualization of edge clustering using a subset of DBLP with 95 instances. Edges are clustered into 10 groups, with each shown in a different color.

representation for his/her true affiliations. Because this edge class information is not readily available in most social media datasets, an unsupervised clustering algorithm can be applied to partition the edges into disjoint sets such that each set represents one potential affiliation [23]. The edges of actors who are involved in multiple group affiliations are likely to be separated into different sets which in turn facilitates the multi-label classification task.

In this paper we construct the node's social feature space using the scalable edge clustering method proposed in [23]. Specifically, we first represent each edge in a feature-based format, where each edge is characterized by its adjacent nodes, as shown in Figure 1. Based on the features of each edge, K-means clustering is used to separate the edges into groups. Each edge cluster represents a potential affiliation, and a node will be considered involved in one affiliation as long as any of its connections are assigned to that affiliation. Since the edge feature data is very sparse, the clustering process can be accelerated wisely. In each iteration a small portion of relevant instances (edges) that share features with cluster centroids are identified, and only the similarity of the centroids with the relevant instance need to be recomputed. By using this procedure introduced by [23], the clustering task can be completed within minutes even for networks with millions of nodes. Figure 2 shows a result of the edge clustering method on a small sample of DBLP dataset. Edges are clustered into 10 separate groups, and each edge group is marked in one color. As we can see, the edge clustering method is able to maintain the correlation between connected nodes; nodes and their neighbors usually share the same type of edge. Also, nodes with high degree are more likely to associate with different types of edges since they are usually involved in multiple affiliations.

After clustering the edges, we can easily construct the node's social feature vector using aggregation operators such as count or proportion on edge cluster IDs. In [23], these *social dimensions* are constructed based on the existence of the node's involvements in different edge clusters. Although aggregation operators are simply different ways of representing the same information (the histogram of edge cluster labels), alternate representations have been shown to impact classification accuracy based on the application domain [20].

3. EXPERIMENTAL SETUP

We evaluate the classification performance of our proposed multi-label relational classifier on three real-world multi-label relational datasets, DBLP, IMDb, and YouTube, whose properties are summarized in Table 2.

Table 2: Dataset Summary

Data	DBLP	IMDb	YouTube
# of nodes	8,865	11,476	15,000
# of links	12,989	323,892	136,218
# of categories	15	27	47
Network Density	3.3×10^{-4}	4.7×10^{-3}	1.2×10^{-3}
Maximum Degree	86	290	14,999
Average Degree	3	55	9
Average Category	2.3	1.5	2.1

3.1 DBLP Dataset

The first real-world dataset we studied in this paper is extracted from the DBLP dataset.[1] The DBLP dataset provides bibliographic information for millions of computer science references. In this paper, we construct a weighted collaboration network for authors who have published at least 2 papers during the 2000 to 2010 timeframe. In this network, the author is represented by the node, and two authors are linked together if they have collaborated at least twice. The weight of the link is defined as the number of times these two authors have co-authored papers. Each author can have multiple research interests. For our dataset, we selected 15 representative conferences in 6 research areas. An author is interested in a research area if he/she has published a paper in any of the conferences listed under that area, and our classification task is to associate each author with the correct set of conferences. The selected conferences are listed below:

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

3.2 IMDb Dataset

The second dataset studied in this paper is IMDb.[2] The Internet Movie Database (IMDb) is an online database of information related to movies, television programs, and video games, including information about directors, actors, and plots. Our classification task is to predict the movie's genres based solely on the collaboration network. Each movie can be assigned to a subset of 27 different candidate movie genres in the database such as "Drama", "Comedy", "Documentary" and "Action". In our experiment, we extract movies and TV shows released between 2000 and 2010, and those directed by the same director are linked together. We only retain movies and TV programs with greater than 5 links.

3.3 YouTube Dataset

The third dataset is extracted from YouTube, which is a popular website for sharing videos. Each user in YouTube can subscribe to different interest groups and add other users as his/her contacts. In this paper, we select a subset of data (15000 nodes) from the original YouTube dataset[3] in [23] using *snowball sampling*, and re-

[1] http://www.informatik.uni-trier.de/~ley/db/
[2] http://www.imdb.com/interfaces
[3] http://www.public.asu.edu/~ltang9/social_dimension.html

tain 47 interest groups as our class label. Unlike DBLP and IMDb, YouTube is not a collaboration network and thus exhibits different network properties.

3.4 Baseline Methods

In this paper, we compare our proposed multi-label relational classifier to four related methods: *EdgeCluster*, *wvRN*, *Prior* and *Random*. A short description of these methods follows:

• *Edge (EdgeCluster)* captures the node's correlation to different classes by extracting social dimensions from network structure using the edge clustering representation [23]. The edge-based social features are constructed using the *count* operator on the edge cluster IDs, and a linear SVM is used as the classifier. To achieve good performance with EdgeCluster, it is necessary to balance the sizes of the positive and negative training sets; this can be accomplished using resampling.

• *wvRN*, weighted-vote Relational Neighbor Classifier [13], makes predictions based solely on the class labels of the given node's linked neighbors; the node's predicted class memberships are constructed as the weighted mean of its neighbors. Our implementation of *wvRN* uses the same relaxation labeling procedure as used in *SCRN*.

• *Prior* generates a class membership estimate according to the fraction of instances in the labeled training data with the given class label. Thus, all nodes (regardless of network connectivity) share the same class estimates which are assigned to multi-label nodes in rank order.

• *Random* generates class membership estimates randomly for each node in the network using neither network nor label information.

In our proposed method, the edge clustering method is initially adopted to construct the social features. We use cosine similarity while performing the clustering; the dimensionality of the edge-based social features is set to 1000 for DBLP and Youtube datasets and 10000 for the IMDb dataset; these parameters are selected because they give the best results for *EdgeCluster* and therefore provide the fairest comparison.

In *SCRN*, the class-propagation probability is calculated by the similarity between the node's social feature and class reference features. We evaluated several similarity measures, including *Cosine*, *Inner Product* and *Generalized Histogram Intersection Kernel*, and we observe that the Generalized Histogram Intersection Kernel (GHI) [2] outperforms the other measures in grouping similar instances and is therefore used in the rest of this paper.

Since our problem is essentially a multi-label classification task, we assume that the number of labels for the unlabeled nodes is already known (e.g., based on the output of a separate classifier) and assign the labels according to the top-ranking set of classes at the conclusion of the inference process. Such a scheme has been adopted for multi-label evaluation in social network datasets [23, 26]. In our work, we sample a small portion of nodes uniformly from the network as training instances. The fraction of the training data ranges from 5% to 30% for DBLP dataset, 1% to 20% for IMDb dataset, and 1% to 9% for the YouTube dataset. We employ the network cross-validation (NCV) method [16] to reduce the overlap between test samples, which produces fair comparisons between different within-network classification approaches. The NCV method starts by creating k disjoint test sets. Then for each test set fold, the remaining folds are merged together, and the training set is randomly sampled from the merged set. The collective inference is executed over the full set of unlabeled nodes (the inference set), but model performance is only be evaluated on the nodes assigned to the test set. The classification performance is evaluated using three standard measures: Macro-F1, Micro-F1, and Hamming Loss. Table 3 summarizes the NCV procedure [16].

Table 3: Network cross-validation procedure

Input: *G*, *propLabeled*, *k*,
S = total number of instances in *G*
F = ∅
Split data into *k* disjoint folds
for *fold* 1 to *k*
  current *fold* becomes *testSet*
  remaining folds are merged to become *trainPool*
  trainSet = sample of (*propLabeled* × *S*) nodes drawn with uniform probability from *trainPool*
  inferenceSet = *G* - *trainSet*
  F = *F* ⋃ <*trainSet, testSet, inferenceSet* >
end for
output: F

3.5 Evaluation Measures

In this section, we explain the details of the evaluation criteria: Macro-F1, Micro-F1 and Hamming Loss. Given the dataset $X \in R^{N \times M}$, let $y_i, \hat{y}_i \in \{0,1\}^K$ be the true and predicted label sets, respectively, for the instance x_i.

• **Macro-F1** [4] is the averaged F1 score over categories:

$$\text{Macro-F1} = \frac{1}{K}\sum_{k=1}^{K} F_1^k. \tag{8}$$

For a category C_k, if P^k and R^k denote the precision and the recall, respectively, Macro-F1 is defined as the harmonic mean of precision and recall:

$$F_1^k = \frac{2P^kR^k}{P^k + R^k} = \frac{2\sum_{i=1}^{N} y_i^k \hat{y}_i^k}{\sum_{i=1}^{N} y_i^k + \sum_{i=1}^{N} \hat{y}_i^k}. \tag{9}$$

• **Micro-F1** [4] is computed using F_1^k while considering the precision as a whole. Specifically, it is defined as follows:

$$\text{Micro-F1} = \frac{2\sum_{k=1}^{k}\sum_{i=1}^{N} y_i^k \hat{y}_i^k}{\sum_{k=1}^{K}\sum_{i=1}^{N} y_i^k + \sum_{k=1}^{K}\sum_{i=1}^{N} \hat{y}_i^k}. \tag{10}$$

Macro-F1 is more sensitive to the performance of rare categories (since all categories are weighted evenly) while Micro-F1 is affected more by the common categories (since this measure weights instances evenly).

• **Hamming Loss** [28] is one of the most frequently used criteria, which counts the number of labels that are incorrectly predicted.

$$\text{HammingLoss} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{K}||y_i \otimes \hat{y}_i||_1, \tag{11}$$

where $\otimes$ denotes the Hamming distance (XOR operation), and $||\cdot||_1$ denotes the $l1$ -norm. The smaller the value, the better the performance of the classifier.

4. RESULTS

We perform two studies to evaluate the performance of our proposed multi-label relational classifier. First, we study the performance of *SCRN* under different measures of calculating the node similarity, $w(v_i, v_j)$. Then we compare the classification results of

Table 4: *SCRN* results using different node similarity measures on DBLP (10% training data)

	Degree	Cosine	Pearson
Micro-F1 (%)	**56.51**	42.96	54.39
Macro-F1 (%)	**49.35**	36.99	47.33

SCRN against four baseline methods on the DBLP, YouTube and IMDb datasets.[4]

4.1 Node Similarity Measures

Both the *wvRN* and *SCRN* classifiers consider the similarity of linked nodes, $w(v_i, v_j)$, when estimating the label of node v_i. $w(v_i, v_j)$ measures the similarity between linked nodes; note that the weight matrix W is not necessarily symmetric (i.e., $w(v_i, v_j)$ can be different from $w(v_j, v_i)$). In this experiment we compare three different approaches for determining the node similarity using the information contained in the network structure.

• *Degree* calculates the weight $w(v_i, v_j)$ by the normalized fraction of connections between v_i and v_j among all of v_i's connections. In our weighted DBLP dataset, we normalize the original weight of the link, $w_0(v_i, v_j)$, by the total weight summed over the neighbors of node v_i, $\sum_{j \in N_i} w_0(v_i, v_j)$, and use it as an estimate of the node's similarity to v_j.

• *Cosine Similarity* uses the cosine function to normalize the number of common neighbors between two nodes in the graph.

• *Pearson Correlation Coefficient* is an alternative way to normalize the count of common neighbors by comparing it with the expected value that the count would have in a network in which nodes select their neighbors at random [19].

Table 4 shows the classification performance of *SCRN* using different node similarity measures. The *Degree* similarity measure clearly achieves the highest accuracy rate (Macro-F1 score of 49.35%); the *Pearson Correlation Coefficient* performs slightly worse than *Degree*; and *Cosine Similarity* is poorest at capturing the relationship between two nodes. Based on this experiment, we select the *Degree* method to measure node similarity for the remaining experiments in the paper.

4.2 Classification Results

Table 5 shows the classification performance, under Macro-F1 and Micro-F1 measures, on the DBLP dataset averaged over 10 cross-validation folds. We make several observations. First, we confirm that all the network classification approaches, which consider the correlations between linked nodes (*SCRN*, *EdgeCluster* and *wvRN*) always outperform the two baseline methods, *Random* and *Prior*. *wvRN*, which takes advantage of the correlation between the labels of linked nodes, significantly outperforms the baselines. *EdgeCluster*, which uses social features in a supervised learning framework, performs worse than *wvRN* on this dataset, since it is less able to exploit label homophily. Our proposed method *SCRN*, which leverages both social features and neighboring labels, consistently outperforms the others. The class-propagation probability in *SCRN* captures the node's intrinsic likelihood of belonging to each class, enabling a more accurate inference procedure.

Table 6 shows results on the IMDb dataset. We observe that *SCRN* consistently has the best performance on Micro-F1. On

[4]Our open-source implementation of *SCRN* and the baseline methods is available at: `http://code.google.com/p/multilabel-classification-on-social-network/`.

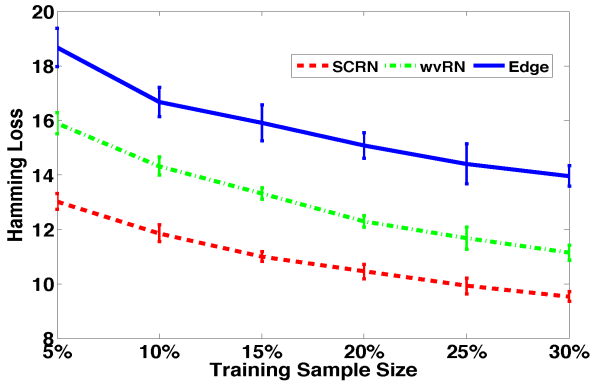

Figure 3: Classification on DBLP Dataset (Hamming Loss); lower score corresponds to better performance. *SCRN* is significantly better.

Macro-F1, *SCRN* and *wvRN* are tied and significantly outperform the non-relational methods. In contrast, we observe that *EdgeCluster* performs surprisingly well on the YouTube dataset, as seen in Table 7. In particular, under the Macro-F1 measure, *SCRN* is outperformed by *EdgeCluster*, although *SCRN* is still the best under the Micro-F1 measure for most conditions. We attribute this to the fact that the YouTube network is not a true collaboration network with strong causal links between authors; also it has a large number (47) of highly skewed classes with a less informative link structure. The relatively low correlation between labels of linked nodes penalizes relational classifiers such as *SCRN* and *wvRN*.

Our results confirm that in multi-label collaborative networks, such as DBLP and IMDb, the correlation between connected nodes can be a great asset for relational learning. However, we argue that it is important to correctly exploit this information. For instance, in previous work by Tang and Liu [22, 24], combining labeled node features and relational features aggregated from neighbors in a link-based classifier performed poorly. Thus, in our proposed approach, rather than simply concatenating these two types of features, we translate the similarity between two connected nodes' social features into a class propagation probability and see that this significantly boosts the performance of collective classification.

Figures 3, 4, and 5 compare the classification performance of the various methods on the DBLP, IMDB, and YouTube datasets, respectively, under the commonly used *Hamming Loss* measure for multi-label classification. *SCRN* significantly outperforms the other methods on both DBLP and IMDb, particularly with fewer training samples. All three methods perform equivalently on the YouTube dataset, as discussed above.

5. RELATED WORK

Multi-label classification (MLC) is a variant of classification where each instance is associated with multiple labels. Given a set of training samples, each of which is associated with a set of labels, MLC aims to learn a model that outputs a bipartition of the labels into those relevant and irrelevant with respect to a query instance. One simple way of addressing multi-label learning is to transform the multi-label classification problem into a set of independent, single-label classification problems, e.g., the most intuitive one-vs-rest learning methods [11]. More sophisticated approaches focus on exploiting the correlations between different labels to improve the label set prediction performance. For instance, Guo and Gu [8]

Table 5: Classification on DBLP Dataset (Macro-F1 and Micro-F1)

Labeled	5%	10%	15%	20%	25%	30%
Micro-F1 (%)						
SCRN	**51.06±1.08**	**56.51±1.18**	**60.31±0.70**	**62.80±0.84**	**65.03±1.13**	**66.58±0.95**
Edge	38.41±2.39	43.65±1.69	47.53±2.54	50.29±1.55	52.50±2.42	54.00±1.26
wvRN	47.59±1.22	52.78±1.29	56.67±0.87	59.45±0.59	61.51±1.36	63.24±0.97
Prior	32.22±1.48	33.06±1.60	32.43±1.85	33.45±1.22	33.23±0.94	33.50±1.04
Random	20.32±0.75	20.65±0.83	20.59±0.97	20.06±0.96	19.84±1.05	20.40±0.78
Macro-F1 (%)						
SCRN	**44.35±1.45**	**49.35±1.51**	**53.65±1.37**	**56.38±1.26**	**58.62±1.21**	**59.54±0.94**
Edge	33.83±1.62	40.26±2.03	43.47±1.14	46.12±1.18	47.36±2.11	49.29±0.98
wvRN	41.85±1.48	46.61±1.49	51.05±1.53	53.88±1.18	55.83±1.70	57.08±1.51
Prior	15.31±1.11	15.76±1.52	15.42±1.41	16.06±0.97	16.13±0.72	16.48±0.81
Random	18.66±0.74	18.97±0.70	18.97±0.91	18.42±0.94	18.20±0.98	18.69±0.67

Table 6: Classification on IMDb Dataset (Macro-F1 and Micro-F1)

Labeled	1%	3%	5%	10%	15%	20%
Micro-F1 (%)						
SCRN	**45.62±2.03**	**58.58±1.39**	**63.65±1.07**	**68.90±1.69**	**71.01±0.70**	**71.98±1.24**
Edge	40.08±1.51	52.17±0.85	57.31±1.56	62.03±1.89	64.50±0.85	65.27±1.32
wvRN	44.72±1.91	56.98±1.35	62.44±1.18	67.05±1.89	70.76±0.92	71.78±1.36
Prior	39.67±1.49	39.48±1.20	39.37±1.23	39.27±1.11	39.28±1.30	39.20±1.14
Random	7.58±0.61	7.23±0.72	7.77±0.60	7.43±0.99	7.38±0.85	7.75±0.59
Macro-F1 (%)						
SCRN	18.46±2.35	27.19±2.31	**33.22±1.39**	**39.40±3.10**	**42.67±2.57**	**43.31±1.66**
Edge	17.64±1.59	24.24±1.83	29.66±2.13	34.17±2.45	36.61±1.59	37.50±1.54
wvRN	**18.53±2.28**	**27.41±2.06**	33.02±1.76	39.08±2.90	42.10±2.54	43.28±2.11
Prior	5.58±0.49	5.57±0.52	5.49±0.46	5.43±0.41	5.40±0.40	5.34±0.42
Random	6.22±0.53	6.04±0.67	6.40±0.61	6.21±0.94	6.24±0.62	6.31±0.58

Table 7: Classification on YouTube Dataset (Macro-F1 and Micro-F1)

Labeled	1%	3%	5%	7%	9%
Micro-F1 (%)					
SCRN	**35.67±3.54**	40.69±3.35	**43.15±1.35**	**43.76±3.11**	43.93±3.27
Edge	35.44±3.89	**40.92±3.87**	41.76±2.60	43.20±3.88	**44.09±2.89**
wvRN	33.18±5.39	40.08±4.23	42.57±2.56	43.40±3.45	43.87±3.90
Prior	34.32±2.74	37.21±1.94	37.24±2.22	37.83±2.38	37.54±2.04
Random	9.77±2.92	9.91±2.56	9.05±2.92	9.52±2.37	9.66±2.69
Macro-F1 (%)					
SCRN	15.20±4.51	21.43±4.98	23.93±3.75	25.84±4.90	26.00±4.28
Edge	**21.64±3.33**	**25.46±4.23**	**26.73±3.67**	**30.08±3.76**	**30.65±4.08**
wvRN	14.80±4.40	23.53±4.99	24.26±4.03	26.54±4.82	27.57±4.27
Prior	10.58±4.80	11.10±4.64	10.78±4.65	11.04±4.53	11.10±4.42
Random	9.07±3.22	9.08±2.70	8.24±3.06	8.65±2.55	8.78±2.91

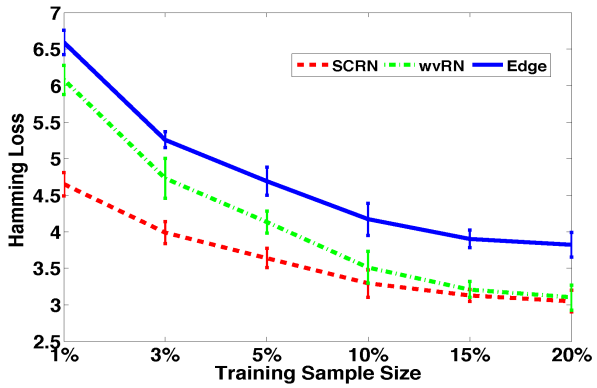

Figure 4: Classification on IMDB Dataset (Hamming Loss); lower score corresponds to better performance. *SCRN* is significantly better.

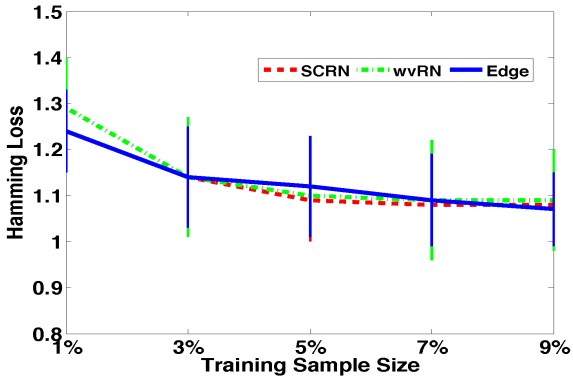

Figure 5: Classification on YouTube Dataset (Hamming Loss); lower score corresponds to better performance. All methods perform equivalently.

proposed a generalized conditional dependency network model for multi-label classification. Their conditional dependency network exploits the dependencies of multiple labels, and the conditional distributions are defined using binary classifiers.

Like other traditional classifiers, MLC assumes that instances are independent and identically distributed. When learning on networked data, relational classifiers can improve on the performance of traditional classifiers by taking advantage of dependencies both among labels, and sometimes among attributes, of related labeled instances [12, 14, 17]. Most of the previous work in collective inference for relational learning uses network connectivity for prediction under the assumption that the connections in the network are homogeneous. However, many real-world networks can be regarded as heterogeneous information networks composed of multiple types of nodes and links. Conventional learning methods do not distinguish the type differences among objects and links in the heterogeneous network. Ji et al. [10] proposed a ranking-based classification model (*RankClass*) for heterogeneous information networks. While classifying the data objects, the model simultaneously ranks each object according to its importance within each class, in order to provide informative class summaries.

Goldberg et al. [7] observe that in social media, nodes may link to one another even if they do not have similar labels. They use two edge types to denote the affinity or disagreement in the class labels of linked objects and incorporate the link type information into discriminant learning. Heatherly et al. [9] introduced a Link Type Relational Bayes Classifier that predicts the node's class labels according to the neighbors' labels as well as their link types. The *SocDim* framework was created specifically to address the link heterogeneity problem [23]. In this framework, latent social dimensions are extracted from the network using modularity maximization to capture the potential affiliations of each entity, and then a discriminant classifier is trained using the instances' social dimensions. Social features were also employed by Wang and Sukthankar [26] in conjunction with Fiedler embedding to uncover the relations between nodes and their links.

6. CONCLUSION

In this paper, we tackled the problem of classifying multi-label networked datasets, where each instance in the network is associated with a subset of multiple labels from the candidate label set. We proposed a multi-label relational classifier (*SCRN*) that addresses the issues that arise when directly applying the relational neighbor classifier (*RN*) on network data.

SCRN combines the ability of relational neighbor classifiers to exploit label homophily while simultaneously leveraging feature similarity through the introduction of class propagation probabilities. Although this paper focuses on the use of social features extracted from the network, it is straightforward to extend our approach to also employ content features constructed from node (e.g., document) properties.

The intuition behind *SCRN* is straightforward: wvRN uses the network solely through class-independent pairwise link strengths during label propagation. In contrast, *SCRN* utilizes an additional observation that strives to capture, on a per class basis, how the given node resembles other nodes based upon the network structure. This observation term thus modifies the probabilities of the node belonging to the different classes. Empirical studies on several real-world tasks demonstrate that our proposed approach significantly boosts classification performance on collaboration networks.

7. ACKNOWLEDGMENTS

This research was supported in part by DARPA award D13AP00002 and NSF IIS-08451.

8. REFERENCES

[1] BHAGAT, S., CORMODE, G., AND MUTHUKRISHNAN, S. Node classification in social networks. *Computing Research Repository (CoRR) abs/1101.3291* (2011).

[2] BOUGHORBELY, S., TAREL, J.-P., AND BOUJEMAA, N. Generalized histogram intersection kernel for image recognition. In *IEEE International Conference on Image Processing* (2005).

[3] CHAKRABARTI, S., DOM, B., , AND INDYK, P. Enhanced hypertext categorization using hyperlinks. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)* (1998), pp. 307–318.

[4] FAN, R., AND LIN, C. A study on threshold selection for multi-label classification. Tech. rep., National Taiwan University, 2007.

[5] FAN, Y., AND SHELTON, C. R. Learning continuous-time social network dynamics. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI)* (2009), pp. 161–168.

[6] GETOOR, L., AND TASKAR, B. *Introduction to Statistical Relational Learning*. The MIT Press, 2007.

[7] GOLDBERG, A., ZHU, X., AND WRIGHT, S. Dissimilarity in graph-based semi-supervised classification. In *Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS)* (2007).

[8] GUO, Y., AND GU, S. Multi-label classification using conditional dependency networks. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)* (2011), pp. 1300–1305.

[9] HEATHERLY, R., KANTARCIOGLU, M., AND LI, X. Social network classification incorporating link type. In *Proceedings of IEEE Intelligence and Security Informatics (ISI)* (2009), pp. 19–24.

[10] JI, M., HAN, J., AND DANILEVSKY, M. Ranking-based classification of heterogeneous information networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2011), pp. 1298–1306.

[11] LEWIS, D. D., YANG, Y., ROSE, T. G., AND LI, F. RCV1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research 5* (Dec 2004), 361–397.

[12] LU, Q., AND GETOOR, L. Link-based classification. In *Proceedings of 20th International Conference on Machine Learning (ICML)* (2003), pp. 496–503.

[13] MACSKASSY, S. A., AND PROVOST, F. A simple relational classifier. In *Proceedings of the Second Workshop on Multi-Relational Data Mining (MRDM) at KDD 2003* (2003), pp. 64–76.

[14] MACSKASSY, S. A., AND PROVOST, F. Classification in networked data: a toolkit and a univariate case study. *Journal of Machine Learning 8* (2007), 935–983.

[15] MCPHERSON, M., SMITH-LOVIN, L., AND COOK, J. M. Birds of a feather: Homophily in social networks. *Annual Review of Sociology 27*, 1 (2001), 415–444.

[16] NEVILLE, J., GALLAGHER, B., ELIASSI-RAD, T., AND WANG, T. Correcting evaluation bias of relational classifiers with network cross validation. *Knowledge and Information Systems* (Jan 2011), 1–25.

[17] NEVILLE, J., AND JENSEN, D. Iterative classification in relational data. In *Proceedings of the AAAI Workshop on Learning Statistical Models from Relational Data* (2000), pp. 42–49.

[18] NEVILLE, J., JENSEN, D., FRIEDLAND, L., AND HAY, M. Learning relational probability trees. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)* (2003), pp. 625–630.

[19] NEWMAN, M. *Networks: An Introduction*. Oxford University Press, 2010.

[20] SEN, P., NAMATA, G., BILGIC, M., GETOOR, L., GALLAGHER, B., AND ELIASSI-RAD, T. Collective classification in network data. *AI Magazine* (2008), 93–106.

[21] SINGH, A., AND GORDON, G. A Bayesian matrix factorization model for relational data. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI)* (2010), pp. 556–563.

[22] TANG, L., AND LIU, H. Relational learning via latent social dimensions. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2009), KDD '09, pp. 817–826.

[23] TANG, L., AND LIU, H. Scalable learning of collective behavior based on sparse social dimensions. In *Proceedings of International Conference on Information and Knowledge Management (CIKM)* (2009).

[24] TANG, L., AND LIU, H. Leveraging social media networks for classification. *Data Mining and Knowledge Discovery (DMKD 2011) 23*, 3 (Nov. 2011), 447–478.

[25] TASKAR, B., ABBEEL, P., AND KOLLER, D. Discriminative probabilistic models for relational data. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)* (2002), pp. 895–902.

[26] WANG, X., AND SUKTHANKAR, G. Extracting social dimensions using Fiedler embedding. In *Proceedings of IEEE International Confernece on Social Computing* (2011), pp. 824–829.

[27] YEDIDIA, J. S., FREEMAN, W. T., AND WEISS, Y. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory 51* (2005), 2282–2312.

[28] ZHANG, X., YUAN, Q., ZHAO, S., FAN, W., ZHENG, W., AND WANG, Z. Multi-label classification without the multi-label cost. In *Proceedings of SIAM International Conference on Data Mining* (Apr. 2010).