Relational Learning via Latent Social Dimensions

Lei Tang
Department of CSE
Arizona State University
Tempe, AZ 85287, USA
L.Tang@asu.edu

Huan Liu
Department of CSE
Arizona State University
Tempe, AZ 85287, USA
Huan.Liu@asu.edu

ABSTRACT

Social media such as blogs, Facebook, Flickr, etc., presents data in a network format rather than classical IID distribution. To address the interdependency among data instances, relational learning has been proposed, and collective inference based on network connectivity is adopted for prediction. However, the connections in social media are often multi-dimensional. An actor can connect to another actor due to different factors, e.g., alumni, colleagues, living in the same city or sharing similar interest, etc. Collective inference normally does not differentiate these connections. In this work, we propose to extract latent social dimensions based on network information first, and then utilize them as features for discriminative learning. These social dimensions describe different affiliations of social actors hidden in the network, and the subsequent discriminative learning can automatically determine which affiliations are better aligned with the class labels. Such a scheme is preferred when multiple diverse relations are associated with the same network. We conduct extensive experiments on social media data (one from a real-world blog site and the other from a popular content sharing site). Our model outperforms representative relational learning methods based on collective inference, especially when few labeled data are available. The sensitivity of this model and its connection to existing methods are also carefully examined.

Categories and Subject Descriptors

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

General Terms

Algorithm, Experimentation

Keywords

Social Dimensions, Behavior Prediction, Social Media

KDD'09, June 28–July 1, 2009, Paris, France.

1. INTRODUCTION

Social media, in forms of Web 2.0 and popular social networking sites like Facebook, Flickr, Youtube, Digg, Blog, etc., is reshaping various fields including online business, marketing, epidemics and intelligent analysis. Concomitant with the opportunities indicated by the rocketing online traffic in social media[1] are the challenges for user/customer profiling, accurate user search, matching, recommendation as well as effective advertising and marketing. Take blogsphere as an example. Bloggers can upload tags for their own blog sites. The tags of a blog site provide the description of the blogger, facilitating blog search, retrieval and other tasks. Unfortunately, not all the bloggers provide tags; even if some do, they may just choose some for convenience. Thus, it becomes a challenge to infer the likely tags of those bloggers with partial information.

Another problem is social networking advertising. Currently, advertising in social media has encountered many challenges[2]. A recent study[3] from the research firm IDC suggested that "just 57% of all users of social networks clicked on an ad in the last year, and only 11% of those clicks lead to a purchase". Note that some social networking sites can only collect very limited user profile information, either due to the privacy issue or because the user declines to share the true information. On the contrary, the friendship network is normally available. If one can leverage a small portion of user information and the network data wisely, the situation might improve significantly.

Both aforementioned two problems can be considered as a classification problem. Consider the tags in blogosphere or user interests as labels, the key task boils down to classifying users into relevant categories. Note that in both cases, some labeled data are readily available: a) In blogosphere, some bloggers do provide accurate and descriptive tags; b) For social networking advertising, online activities of users such as clicking on an ad or purchasing a product reflect the users' potential interests. In reality, this kind of label information is very limited compared with the whole user population. Different from conventional data mining where data are assumed to be independently identically distributed (IID), the data here (specifically, the social actors) are connected in a network. So the key problem is *how can we leverage the*

[1]http://www.techcrunch.com/2008/12/31/top-social-media-sites-of-2008-facebook-still-rising/
[2]http://www.nytimes.com/2008/12/14/business/media/14digi.html
[3]http://www.nytimes.com/2008/12/01/technology/internet/01facebook.html

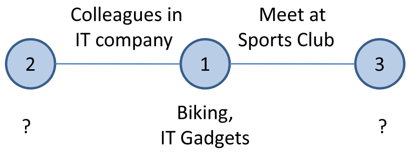

Figure 1: Toy Example

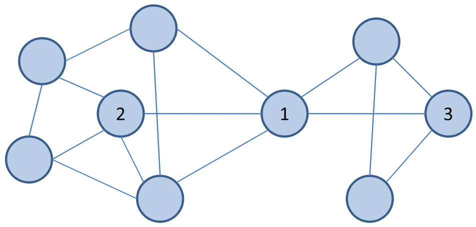

Figure 2: A Snapshot of Node 1's local Network

social network information for accurate classification when limited label information is available?

Social networks do provide some valuable information through homophily [21]. "Birds of a feather flock together." It is observed that similarity breeds connections in real-world social networks. In terms of social media, actors sharing some interests tend to interact with each other, thus leading to autocorrelations between the labels of connected actors. Relational learning (within-network classification) [20, 10] has been proposed specifically to capture the correlations between connected objects. Specifically, if there is only one network among data objects, one basic Markov assumption [20] is that the label of one node is dependent on that of its neighbors. For prediction, collective inference is required to find an equilibrium status such that the inconsistency between neighboring nodes in the network is minimized. This is normally done by iteratively updating labels or class membership of one node while fixing the other nodes in the network. Relational learning with collective inference has been shown to outperform models that do not consider the connectivity information [4, 32, 23].

One limitation of collective inference models is that they treat the connections in the network homogeneously. In the real world, various reasons lead to heterogeneous connections. People connect to each other because they are colleagues, classmates, friends, or share similar interest or political views. Currently, most collective inference models do not differentiate the connections between actors and this might blur the class membership of actors in the network. To give a palpable understanding, let us look at a toy example in Figure 1. Actor 1 connects to Actor 2 because they work in the same IT company, and connects to Actor 3 because they often meet each other in the same sports club. Given the label information that Actor 1 is interested in both Biking and IT Gadgets, can we infer Actor 2 and 3's labels? Treating these two connections homogeneously, we guess that both Actor 2 and 3 are also interested in biking and IT gadgets. But if we know how Actor 1 connects to other actors, it is more reasonable to conjecture that Actor 2 is more interested in IT gadgets and Actor 3 likes biking.

The above example is ideal since the cause of connections is explicitly known. But this kind of information is rare in real-world applications, though some social networking sites like Facebook do ask how two get to know each other when users add a friend. Most of the time, only the network connectivity (as in Figure 2) is available. If we can

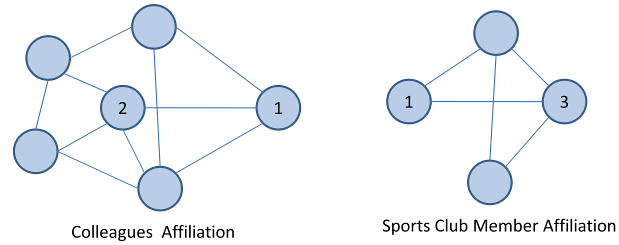

Figure 3: Different Affiliations with Node 1

somehow differentiate the connections into different affiliations as shown in Figure 3, then we can possibly infer the class membership of each actor more precisely. Notice that Actor 1 is presented in multiple different affiliations. This is consistent with the multi-facet property of human nature.

Given limited information and the network connectivity, differentiating the connections into different affiliations is by no means an easy task as the same actor is involved in multiple affiliations. Moreover, the same connection can be associated with multiple affiliations. For instance, one can connect to another as they are colleagues and also go to the same sports club frequently. Instead of capturing affiliations among actors via differentiating connections directly, we resort to latent social dimensions, with each dimension representing a plausible affiliation among social actors.

In this work, we present a relational learning framework based on latent social dimensions. Each dimension can be considered as the description of a likely affiliation between social actors. With these social dimensions, we can harness the power of discriminative learning such as SVM or logistic regression to automatically select the relevant social dimensions for classification. In the prediction phase, different from existing relational learning methods, collective inference becomes unnecessary as the selected social dimensions have already included the relevant network connectivity information. This proposed framework is flexible and allows for the combination of other features such as user profiles or social content information.

Different from existing relational learning works, which often concentrate on entity resolution, web page or publication classification, we specifically focus on classification associated with social media where the network is noisy and typically has a composite of multiple relations among actors. We also study a more complicated situation that each actor can belong to multiple class labels instead of just one single label [20]. Extensive experiments were conducted on two data sets: one is collected from a real-world blog site and the other from a popular content sharing site. It is demonstrated that network classification with latent social dimensions outperforms alternative relational learning methods. In addition, we show that latent social dimensions, in conjunction with other features like content or tags extracted from social media, can improve the performance significantly.

2. PROBLEM FORMULATION

In social media, individuals are highly idiosyncratic. Each actor's interest cannot be captured by mere one class. Take Flickr[4] as an example. One user in Flickr subscribes to as many as 4344 interest groups[5]. Rather than concentrating

[4] http://www.flickr.com/

[5] http://www.flickr.com/people/8551473@N08/

on univariate cases for classification in network data [20] (each node has only one class label), here we examine more challenging tasks that each node in a network can have multiple labels. The problem we study can be formally described below:

> Suppose there are K class labels $\mathcal{Y} = \{c_1, \cdots, c_K\}$. Given network $\mathcal{A} = (V, E, Y)$ where V is the vertex set, E is the edge set and $Y_i \subseteq \mathcal{Y}$ are the class labels of a vertex $v_i \in V$, and given known values of Y_i for some subsets of vertices V^L, how to infer the values of Y_i (or a probability estimation score over each label) for the remaining vertices $V^U = V - V^L$?

For convenience, we use $\mathbf{y}_i \in \{0, 1\}^K$ to denote the class labels associated with node v_i. Typically, relational learning makes the following Markov assumption:

$$P(\mathbf{y}_i|\mathcal{A}) = P(\mathbf{y}_i|\mathcal{N}_i) \quad (1)$$

where $\mathcal{N}_i$ is a set of "neighbors" of vertex v_i. The neighbors are normally defined as vertices that are 1-hop or 2-hop away from v_i in the network [14, 8]. A relational classifier based upon the labels of neighbors can be learned via the labeled vertices V^L. For prediction, the labels of unlabeled vertices are initialized. Then the constructed relational classifier assigns class labels or update class membership for each node while fixing labels of the other nodes. This process is repeated until convergence. Relaxation labeling [4], iterative classification [18] and Gibbs sampling [9] are the commonly used techniques. Please refer to [20] for details.

The assumption in Eq. (1) is essentially local. It does not capture the weak dependencies between nodes that are not close or directly connected. Expanding the neighbor set can possibly increase the capacity of modeling complicated dependency if over-fitting is not an concern, but it requires much more computational efforts for the convergence of subsequent collective inference, as the small-world phenomenon [33] is almost universally observed in social networks. Expanding the neighbor set to include nodes that are several hops away would include a large portion of the whole network.

Moreover, a social network is often a composite of various relations. One user can connect to his/her friends, alumni, colleagues or family members. He can also connect to other virtual friends if they share some interesting topics. The diversity of connections does not necessarily indicate that two connected users would share certain class label. Relying network alone for collective inference does not distinguish these connections. It becomes a challenge to detect which connections are informative for one class label. This is analogous to the feature selection problem in classical data mining if we were lucky enough to have these connection types. But in reality, this kind of information is rare or not refined enough for to be informative. Thereafter, we aim to identify latent social dimensions which are informative of affiliations among actors. In these dimensions, the weak dependency among "distant" actors can also be captured. Next, we shall illustrate some principles to extract latent social dimensions and present an algorithm of *discriminative relational learning*.

3. LATENT SOCIAL DIMENSIONS

The social dimensions extracted from the network should satisfy the following properties:

- Informative. The social dimensions should be indicative of affiliations between actors.
- Plural. The same social actor can get involved in multiple affiliations, thus appearing in different social dimensions.
- Continuous. The actors might have different degree of associations to one affiliation. Hence, a continuous value rather than discrete $\{0, 1\}$ is more favorable.

As introduced above, we aim to extract the social dimensions that are indicative of affiliations between actors. Based on homophily[21], similar actors interact at higher rate than those dissimilar ones. Thus, actors sharing certain properties tend to form groups with denser within-group interactions. This naturally connects to one basic task in social network analysis — community detection, which aims to find out communities that have denser within-group interaction than between-group interactions. While most community detection algorithms partition the actors into several disjoint clusters, we allow the same actor to be involved in different affiliations. Hence, a soft clustering method is adopted.

Many approaches have been developed for clustering on graphs that serve the purpose of social dimension extraction, including graph partitioning [15], latent space model [12, 28], block model [27, 1], spectral clustering [36], etc. In large scale social networks, scale free property [3] is commonly observed. In other words, the degree of nodes in a network follows a power law distribution. Modularity [25] is a recently proposed community measure that explicitly takes the degree distribution into consideration and has been shown to be an effective quantity to measure community structure in many complex networks [6].

Here, we briefly review the concept of modularity. Consider dividing the interaction matrix A of n vertices and m edges into k non-overlapping communities. Let s_i denote the community membership of vertex v_i, d_i represents the degree of vertex i. Modularity is like a statistical test that the null model is a uniform random graph model, in which one actor connects to others with uniform probability. For two nodes with degree d_i and d_j respectively, the expected number of edges between the two in a uniform random graph model is $d_i d_j / 2m$. Modularity measures how far the interaction deviates from a uniform random graph wth the same degree distribution. It is defined as:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{d_i d_j}{2m} \right] \delta(s_i, s_j) \quad (2)$$

where $\delta(s_i, s_j) = 1$ if $s_i = s_j$. A larger modularity indicates denser within-group interaction. Note that Q could be negative if the vertices are split into bad clusters. $Q > 0$ indicates the clustering captures some degree of community structure. In general, one aims to find a community structure such that Q is maximized.

While maximizing the modularity over hard clustering is proved to be NP hard [2], a relaxation of the problem can be solved efficiently [24]. Let $\mathbf{d} \in Z_+^n$ be the degree of each node, $S \in \{0, 1\}^{n \times k}$ be a community indicator matrix defined below:

$$S_{ij} = \begin{cases} 1 & \text{if vertex } i \text{ belongs to community } j \\ 0 & \text{otherwise} \end{cases}$$

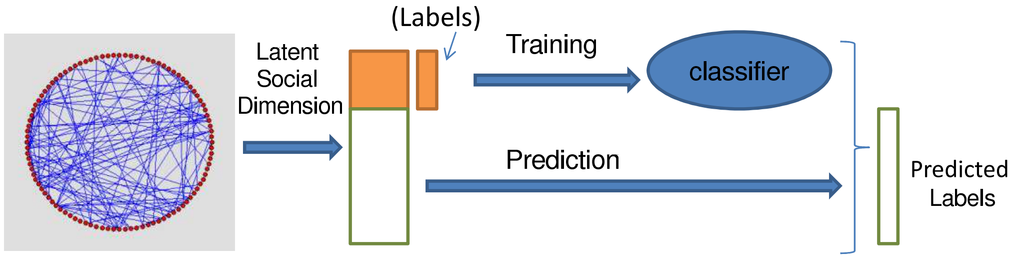

Figure 4: Relational Learning via Latent Social Dimensions

and the modularity matrix defined as

$$B = A - \frac{\mathbf{d}\mathbf{d}^T}{2m} \quad (3)$$

The modularity can be reformulated as

$$Q = \frac{1}{2m} Tr(S^T B S) \quad (4)$$

Relaxing S to be continuous, it can be shown that the optimal S is the top-k eigenvectors of the modularity matrix [24].

While the interactions matrix A is normally very sparse, the modularity matrix B is dense and cannot be computed and held in memory if n is large (which is typically true for real-world social networks). We could use the power method or Lanzcos method to calculate the top eigenvectors, as it relies only on the basic operation of matrix-vector multiplication. As the modularity matrix B is the difference of a sparse matrix A and rank-one matrix $\mathbf{d}\mathbf{d}^T/2m$, we can calculate the multiplication of B and a vector $\mathbf{x}$ as

$$B\mathbf{x} = A\mathbf{x} - \frac{(\mathbf{d}^T\mathbf{x})}{2m}\mathbf{d}$$

With a simplified matrix vector multiplication without explicitly representing the whole modularity matrix, we are able to calculate the top eigenvectors efficiently for large scale networks using existing numerical software packages.

4. ALGORITHM

In this section, we illustrate the detailed procedure of social-dimension-based method for discriminative relational learning. The overall process is shown in Figure 4, which consists of two steps:

1. *Extract latent social dimensions based on network connectivity.* In this work, we focus on modularity. The dimensions can be extracted via the top eigenvectors of the modularity matrix B defined in Eq (3). Other clustering approaches can also be explored as discussed in previous section. Note that the real-world network is very noisy thus we only keep those top representative ones. This also reduces the computational cost of large-scale eigenvector calculation. Since labeled and unlabeled nodes both are involved in the calculation, the latent social dimensions are available for all the nodes after calculation.

2. *Construct Discriminative Classifier.* After we extract the social dimensions, we consider them as normal features and conduct supervised learning. Any classifier like SVM or logistic regression can be used. If some other features are available such as user profile or blog content information, they can also be included during discriminative learning. This step is critical as the classifier will determine which dimensions are relevant to a class label. In our case, we pick one-vs-rest linear SVM due to its simplicity and scalability [31]. More powerful methods like structural SVM [34] can also be applied. Prediction is easy once the classifier is ready, since the latent social dimensions have been calculated for unlabeled data in Step 1. Note that collective inference is not required for prediction.

One concern with modularity maximization is that the obtained features are not unique. Let S be the extracted features based on Eq (4), and P be an orthonormal matrix such that $P \in R^{k\times k}, P^TP = PP^T = I_k$. It can be verified that $S' = SP$ also maximizes the modularity as:

$$\begin{aligned} \frac{1}{2m} tr\left((S')^T B (S')\right) &= \frac{1}{2m} tr\left((SP)^T B (SP)\right) \\ &= \frac{1}{2m} tr\left(S^T BSPP^T\right) \\ &= \frac{1}{2m} tr\left(S^T BS\right) = Q \end{aligned}$$

But this does not affect the discriminative learning if a linear SVM is employed. Linear SVM with social dimensions S can be considered as a kernel machine with a linear kernel $\mathcal{K} = SS^T$. With an orthogonal transformation P, the new kernel $\mathcal{K}'$ does not change since:

$$\mathcal{K}' = S'S'^T = (SP)(SP)^T = SPP^TS^T = SS^T = \mathcal{K}$$

Hence the classifier and the prediction is not affected by the non-uniqueness of Step 1.

5. EXPERIMENTAL SETUP

In this section, we describe the data we collected for experiments and the baseline methods for comparison.

5.1 Data Sets

In this work, we focus on social media. We shall examine how relational learning behaves on real-world social networks. Two data sets are collected: one from BlogCatalog[6] and the other from a popular photo sharing site Flickr[7].

BlogCatalog A blog in BlogCatalog is associated with various information pieces like the categories the blog is listed under, blog level tags, snippets of 5 most recent blog posts, and blog post level tags. Bloggers submit their blogs to BlogCatalog and specify the metadata mentioned above for improved access to their blogs. This way the blog sites

[6] http://www.blogcatalog.com/
[7] http://www.flickr.com/

Table 1: Statistics of Social Network Data

Data	BlogCatalog	Flickr
Categories (k)	39	195
Actors (n)	10, 312	80, 513
Links (m)	333, 983	5, 899, 882
Density	6.3×10^{-3}	1.8×10^{-3}
Maximum Degree	3, 992	5,706
Average Degree	65	146
Average Labels	1.4	1.3

are organized under pre-specified categories. A blogger also specifies his social network of other bloggers. A blogger's interests could be gauged by the categories he publishes his blogs in. Each blogger could list his blog under more than one categories. We pick 39 categories with a reasonably large blogger pool for evaluation purpose. On average each blogger lists their blog under 1.6 categories.

Flickr It is a popular website to host personal photos uploaded by users and also an online community platform. Users in Flickr can tag photos and add contacts. Users can also subscribe to different interest groups ranging from *black and white photos*[8] to a specific subject (say *bacon*[9]). In our experiments, we randomly pick 195 interest groups as the class labels and crawl the contact network among the users subscribed to these groups. The users with only one single connection are removed from the data set.

Table 1 lists some statistics of the network data. As seen in the table, the connection among the social actors are extremely sparse. The degree distribution is highly imbalanced, a typical phenomenon in scale-free networks. Both data sets are available from the first author's homepage.

5.2 Baseline Methods

We compare our proposed method to some representative relational learning methods.

- Latent Social Dimensions Approach (SocDim). We set the number of latent social dimensions to 500 and use one-vs-rest linear SVM for discriminative learning.

- Weighted-Vote Relational Neighbor Classifier (wvRN) [19]. This classifier is like a lazy learner. In prediction, the relational classifier estimates the class membership $p(\mathbf{y}_i|\mathcal{N}_i)$ as the weighted mean of its neighbors.

$$p(\mathbf{y}_i|\mathcal{N}_i) = \frac{1}{Z}\sum_{v_j \in \mathcal{N}_i} w_{ij} p(\mathbf{y}_j|\mathcal{N}_j) \quad (5)$$

$$= \frac{1}{|\mathcal{N}_i|} p(\mathbf{y}_j|\mathcal{N}_j) \quad (6)$$

  where Z in Eq. (5) is a normalization factor. Eq. (6) is derived, as the networks studied use $\{0, 1\}$ to represent connections between actors and we only consider the first order Markov assumption (The labels of one actor depend on his connected friends). wvRN has been shown to work reasonably well for classification in univariate case and is recommended as a baseline method for comparison of relational learning [20].

- Network Only Link-Based Classifier (LBC) [18]. This classifier creates relational features of one node by aggregating the label information of its neighbors. Then a relational classifier can be constructed based on labeled data. Specifically, we use averaged class membership (as in Eq. (6)) as relational features and SVM is adopted for building the relational classifier. For prediction, relaxation labeling is employed.

- Latent Group Classifier (LGC). This classifier follows the idea presented in latent group model [22]. But differently, clustering memberships are used as features for SVM learning. We adopt a similar strategy as in [13] by considering the connections of each user as features and using k-means with cosine similarity for clustering. The number of clusters is set the same as the number of labels. To make a fair comparison, we also tried LGC with the number of latent groups set to be the same as SocDim (Denoted as LGC500).

- Majority Model (MAJORITY). This baseline method uses the label information only. It does not leverage any network information for learning or inference. It simply predicts the class membership as the proportion of positive instances in the labeled data. All nodes are assigned with the same class membership.

- Random Model (RANDOM). As indicated by the name, this model predicts with neither network nor label information. It generates a class membership estimation randomly for each node in the network.

In our experiments, actors might have more than one label. Since most methods yield a ranking of labels rather than an exact assignment, a thresholding process is normally required. It has been shown that different thresholding strategies lead to quite different performance [7, 31]. To avoid the affection of thresholding, we assume the number of labels on the test data are already known and check how the top-ranking predictions match with the true labels. Two commonly used measures Micro-F1 and Macro-F1 are adopted to evaluate the classification performance.

6. EXPERIMENTS

In this section, we examine the performance of different relational learning methods. We also investigate whether collective inference is necessary for SocDim and its sensitivity to the latent dimensionality.

6.1 Performance on BlogCatalog Data

Table 2 presents the performance of various approaches for the BlogCatalog data. We gradually increase the number of labeled nodes from 10% to 90%. For each setting, we randomly sample a portion of nodes as labeled. This process is repeated 10 times and the average results are reported. Bold face in the table denotes the highest performance in each column. Clearly, our proposed SocDim outperforms all the other methods. wvRN, as shown in the table, is the runner-up most of the time. MAJORITY performs even worse than RANDOM in terms of Macro-F1 as it always picks the majority class for prediction.

The performance differences between SocDim and other relational learning methods with collective inference, are plotted in Figure 5. As shown in the figure, the link based classifier (LBC) performs poorly with few labeled data. This is because LBC requires to learn a relational classifier on labeled data before the inference. When samples are few, the

[8] http://www.flickr.com/groups/blackandwhite/
[9] http://www.flickr.com/groups/everythingsbetterwithbacon/

Table 2: **Performance on BlogCatalog Network with 10, 312 nodes**

	Training Ratio	10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1(%)	SocDim	**27.35**	**30.74**	**31.77**	**32.97**	**34.09**	**36.13**	**36.08**	**37.23**	**38.18**
	wvRN	19.51	24.34	25.62	28.82	30.37	31.81	32.19	33.33	34.28
	LBC	13.52	18.88	21.36	25.31	29.44	31.65	32.57	33.77	35.31
	LGC	18.29	19.14	20.01	19.80	20.81	20.86	20.53	20.74	20.78
	LGC500	16.61	18.00	18.87	19.54	20.40	20.65	21.06	21.54	22.04
	MAJORITY	16.51	16.66	16.61	16.70	16.91	16.99	16.92	16.49	17.26
	RANDOM	4.84	4.75	4.88	4.91	4.85	4.91	4.84	4.87	5.05
Macro-F1(%)	SocDim	**17.36**	**20.00**	**20.80**	**21.85**	**22.65**	**23.41**	**23.89**	**24.20**	**24.97**
	wvRN	6.25	10.13	11.64	14.24	15.86	17.18	17.98	18.86	19.57
	LBC	3.34	5.78	7.67	10.53	12.91	14.86	16.02	16.94	17.94
	LGC	7.38	7.02	7.27	6.85	7.57	7.27	6.88	7.04	6.83
	LGC500	8.57	9.11	9.51	9.54	10.07	10.38	10.36	10.32	10.89
	MAJORITY	2.52	2.55	2.52	2.58	2.58	2.63	2.61	2.48	2.62
	RANDOM	4.14	4.03	4.17	4.18	4.11	4.12	4.15	4.10	4.27

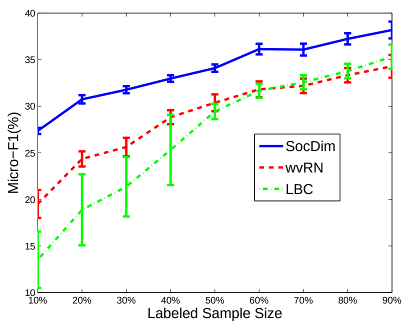

Figure 5: SocDim vs. Collective Inference

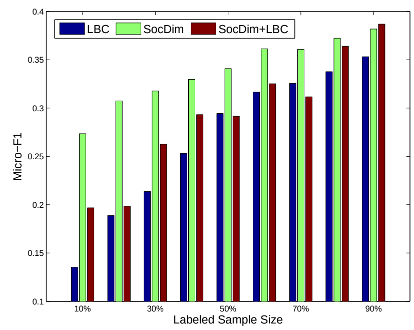

Figure 8: Collective Inference's Effect on SocDim

learned classifier is not stable and robust enough. This is indicated by the large deviation of LBC in the figure when labeled samples are less than 50%. wvRN is more stable, but its performance is not comparable to SocDim. Even with 90% of nodes being labeled, a significant difference between these two models is still observed.

Meanwhile, the latent group classifier (LGC) performs not so well on BlogCatalog. To make a fair comparison, we also include the case (LGC500) such that the number of clusters is the same as the latent dimensionality (500) in SocDim. But increasing the number of clusters does not affect much. SocDim allows the same actor to appear at different latent dimensions, whereas LGC forces each actor to be assigned to only one group. So even with the same discriminative learning procedure, SocDim performs much better than LGC.

6.2 Performance on Flickr Data

Compared with BlogCatalog, Flickr data is in a larger scale with close to 100,000 nodes. In reality, the label information in large scale networks is very limited. Here we examine a similar case with few labeled data. We change the training ratio from 1% to 10%. Roughly, the number of labeled actors increases from around 1000 to 10, 000. The results are reported in Table 3.

It is evident that our SocDim outperforms the other methods almost all the time. Different from BlogCatalog data, LGC is a close runner-up this time. This is because the network is very noisy, as indicated by the low performance of all the methods. Clustering essentially helps remove such noise and keeps prominent patterns. Due to its discrete property, the performance is inferior to SocDim. The other relational learning methods such as wvRN and LBC perform poorly. The LBC fails most of the time (almost like random) and is highly unstable. This can be verified by the fluctuation of Micro-F1 of LBC. Here, we want to reemphasize that the interactions in real-world networks are highly diverse. Detecting the relevant ones from 195 labels is not an easy task. While alternative relational learning methods fail, SocDim works significantly better than those "dummy" methods like RANDOM and MAJORITY.

6.3 Collective Inference + SocDim

In previous sections, SocDim predicts the labels without collective inference. One natural question is whether collective inference can boost the performance of SocDim? Since wvRN does not build a relational classifier, it becomes tricky to combine social dimensions. Hence, we implement a variant of link based classifier with relaxation labeling. The difference is that a relational classifier is learned based on the combination of the labeled nodes' social features and their relational features aggregated from their neighbors.

Figure 8 shows the performance after we combine SocDim with collective inference. Clearly, the social features do help for inference, but collective inference degenerates the performance of SocDim instead of improving it. This is consistent with our conjecture that latent social dimensions have already encoded the necessary network information. Col-

Table 3: Performance on Flickr Network with 80, 513 nodes

	Training Ratio	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
Micro-F1(%)	SocDim	22.75	**25.29**	**27.30**	**27.60**	**28.05**	**29.33**	**29.43**	**28.89**	**29.17**	**29.20**
	wvRN	17.70	14.43	15.72	20.97	19.83	19.42	19.22	21.25	22.51	22.73
	LBC	0.89	0.64	0.32	0.64	15.60	15.64	0.20	15.74	17.73	0.31
	LGC	**22.94**	24.09	25.42	26.43	27.53	28.18	28.32	28.58	28.70	28.93
	LGC500	19.28	21.54	22.39	24.35	25.63	26.81	26.67	27.45	27.76	28.13
	MAJORITY	16.34	16.31	16.34	16.46	16.65	16.44	16.38	16.62	16.67	16.71
	RANDOM	0.94	0.92	0.96	0.92	0.92	0.92	0.93	1.01	0.91	0.93
Macro-F1(%)	SocDim	**10.21**	**13.37**	**15.24**	**15.11**	**16.14**	**16.64**	**17.02**	**17.10**	**17.14**	**17.12**
	wvRN	1.53	2.46	2.91	3.47	4.95	5.56	5.82	6.59	8.00	7.26
	LBC	0.21	0.09	0.11	0.15	1.82	1.57	0.07	1.13	2.47	0.10
	LGC	7.90	9.99	11.42	11.10	12.33	12.29	12.58	13.26	12.79	12.77
	LGC500	7.61	11.53	11.81	13.35	14.38	14.64	15.26	15.16	15.38	15.36
	MAJORITY	0.45	0.44	0.45	0.46	0.47	0.44	0.45	0.47	0.47	0.47
	RANDOM	0.72	0.71	0.75	0.71	0.72	0.70	0.69	0.77	0.71	0.71

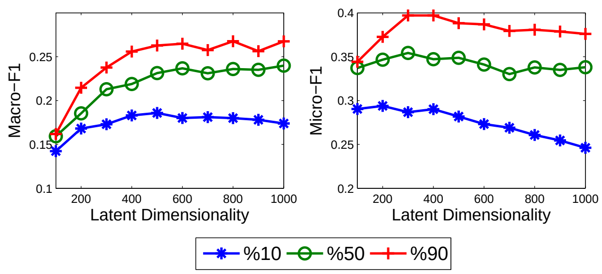

Figure 6: Sensitivity Study on BlogCatalog

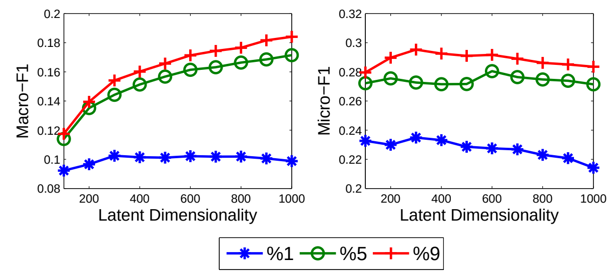

Figure 7: Sensitivity Study on Flickr

lective inference enforces local dependency, while extracted social dimensions alone are capable of capturing the local and weak distant dependency. Therefore, collective inference becomes unnecessary and we can use social dimensions as features for direct prediction.

6.4 Sensitivity of Latent Dimensionality

In previous experiments, we fix the latent dimensionality to 500 for SocDim. In this section, we examine how the performance is affected by the selected number of latent social dimensions. On both data sets, we vary the dimensionality from 100 to 1000 and observe its performance variation. The performance changes on BlogCatalog and Flickr are plotted in Figures 6 and 7 respectively. To make the figures legible, we only plot the cases when 10%, 50% or 90% of nodes in the network are labeled on BlogCatalog and 1%, 5% or 9% of labeled nodes on Flickr.

As seen in the figure, Macro-F1 stabilizes on BlogCatalog after 500 dimensions but increases steadily on Flickr Data. If few (<300) latent social dimensions are selected, then it might miss some discriminative dimensions thus the performance deteriorates. A surprising pattern observed on both data sets is that Micro-F1 decreases with increasing latent dimensions. It seems that the dimensionality is a trade-off between Micro-F1 and Macro-F1. We suggest $400 - 600$ dimensions as a proper range to use SocDim.

6.5 Combination with Other Features

One nice property of SocDim is that it is feature based. Thus, if information is available about the nodes in the network (e.g. user profile , social content or tag information), it is easy to couple the network information and user information: simply combine the extracted social dimensions with other features, and let the discriminative learning procedure to determine which feature are more informative of a class label. The combination of network and actors' features may lead to more accurate classification performance. Here we take BlogCatalog as an example to show the effect.

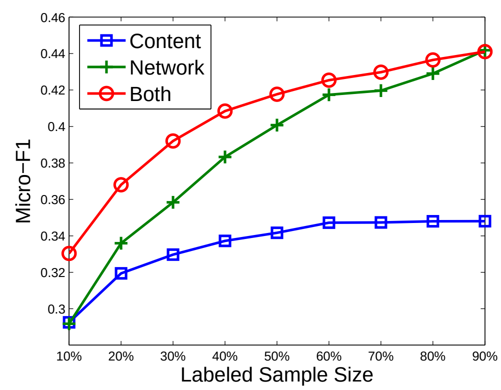

Figure 9: Network + Content in BlogCatalog

BlogCatalog provides the snippets of 5 recent posts of bloggers. We use the snippets as content information about the bloggers. The performances of using content or network alone, or the combination of the two are plotted in Figure 9. As the snippets are short and noisy, it is not surprising that the performance based on content alone is even worse than network based approach. If we combine the social features and the content features, the performance is increased 2-3%. This is most observable when the labeled data are few.

7. RELATED WORK

Relational Learning [10] refers to the classification when objects or entities are presented in multiple relations. In our work, we study classification in network data [20]. The data instances in the network are not independently identically distributed (IID) as in conventional data mining. In order to capture the autocorrelation between labels of neighboring data objects, a Markov dependency assumption is forced. That is, the labels of one node depend on the labels (or attributes) of its neighbors. Collective inference [14] is proposed for prediction. Normally, a relational classifier is constructed based on the relational features of labeled data, and then an iterative process is required to determine the class labels for the unlabeled data. It is shown that a simple weighted vote relational neighborhood classifier [19] works reasonably well on some benchmark relational data and is recommended as a baseline for comparison [20]. It turns out that this method is closely related to Gaussian field for semi-supervised learning on graphs [38].

Most relational classifiers only captures the local dependency based on the Markov assumption. To capture the long-distance autocorrelation, latent group model [22], and nonparametric infinite hidden relational model [37] assume generative models such that the link (and actor attributes) are generated based on the actors' latent cluster membership. But the complexity and high computational cost for inference hinders its direct application to large networks. So Neville and Jensen in [22] propose to use clustering algorithm to find out the hard cluster membership of each actor first, and then fix the latent group variables for later inference. In social media, the network is very noisy. Some nodes do not show a strong community membership and hard clustering might assign them randomly [13]. The resultant community structure can change drastically even with the removal of one single edge in the network. Our latent social dimensions are represented as continuous values and allow each node to involve at different dimensions in a flexible degree. Conjunction with the discriminative power of SVM, this yields a more accurate and stable performance as verified in the experiments.

Another related field is semi-supervised learning. Some works [17, 5] attempt to address semi-supervised learning with multiple labels by utilizing the relationship between different labels. The relationship can be obtained either from external experts or computed based on the labeled data. But its computational cost is prohibitive. We tried the method presented in [5], which constructs a graph between different labels and then find out an label assignment such that it is smooth on both the instance and the label graph. It requires to solve a Sylvester equation and direct implementation takes extremely long time to reach a solution, preventing us from reporting any comparative results.

On the other hand, some papers try to construct kernels based on graphs for SVM. Diffusion kernel[16] is a commonly used one. Unfortunately, it requires full SVD of the graph Laplacian, which is not applicable for large-scale networks. Empirically, the classification performance is sensitive to the diffusion parameter. Cross validation or some variants of kernel learning procedure is required to select a proper diffusion kernel [35].

Community detection has been an active field in social network analysis and various methods has been proposed including stochastic block model [27, 1], latent space model [12, 11], spectral clustering [36], hierarchical clustering based on various measures such as shortest-path betweenness [26] or modularity [25, 24]. In our proposed model, community detection could be used as the procedure to extract latent social dimensions. In this work, we adopt modularity as it is specifically designed for social networks. But any other soft clustering methods serve the purpose as well.

8. CONCLUSIONS AND FUTURE WORK

Social media provides a virtual social networking environment. The classical IID assumption of data instances is not applicable. Relational learning based on collective inference has been proposed to capture the local dependency of labels between neighboring nodes. However, it treats the connections within the network homogeneously. In reality, the connections within the same network are often multi-dimensional. To capture different affiliations among actors in a network, we propose to extract latent social dimensions via modularity maximization. Based on the extracted social features, a discriminative classifier like SVM can be constructed to determine which dimensions are informative for classification. Extensive experiments on social media data demonstrated that our proposed social dimension approach outperforms alternative relational learning methods, especially when the labeled data are few. It is noticed that some relational learning models perform poorly in social media data. This is partly due to the multi-dimensionality of connections and high irregularity of human interactions as presented in social media. Our approach, by differentiating the connections among social actors, achieves effective learning.

In our current model, the obtained social dimensions are orthogonal to each other. We feel that this orthogonality is not a strictly required component. We are currently investigating fast sparse approximation of social dimensions to avoid the large-scale eigenvalue problem. Sometimes, group profiles [29] are available like the group size, connection density and shared topics and attributes. How to utilize this group-level information with social dimensions needs more exploration. Another challenge raised in social media is that the network is highly dynamic and might consists of multiple entities [30]. Each day, new members join the social network, and new connections occur among existing members. How to efficiently update the relational model in this large scale remains a challenge.

9. ACKNOWLEDGMENTS

This research is sponsored by the Air Force Office of Scientific Research grant FA95500810132. We thank BlogCatalog and Flickr for providing APIs. We also acknowledge Xufei Wang for crawling the data in BlogCatalog and Munmun De Choudhury for her help on Flickr crawling.

10. REFERENCES

[1] E. Airodi, D. Blei, S. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, 2008.

[2] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner. Maximizing modularity is hard. *Arxiv preprint physics/0608255*, 2006.

[3] D. Chakrabarti and C. Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM Comput. Surv.*, 38(1):2, 2006.

[4] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *Proceedings of the ACM SIGMOD international conference on Management of data*, 1998.

[5] G. Chen, F. Wang, and C. Zhang. Semi-supervised multi-label learning by solving a sylvester equation. In *SDM*, 2008.

[6] L. Danon, J. Duch, A. Arenas, and A. Díaz-guilera. Comparing community structure identification. *J. Stat. Mech*, 2005.

[7] R.-E. Fan and C.-J. Lin. A study on threshold selection for multi-label classication. 2007.

[8] B. Gallagher, H. Tong, T. Eliassi-Rad, and C. Faloutsos. Using ghost edges for classification in sparsely labeled networks. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008.

[9] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. pages 452–472, 1990.

[10] L. Getoor and B. Taskar, editors. *Introduction to Statistical Relational Learning*. The MIT Press, 2007.

[11] M. S. Handcock, A. E. Raftery, and J. M. Tantrum. Model-based clustering for social networks. *Journal Of The Royal Statistical Society Series A*, 2007.

[12] P. D. Hoff and M. S. H. Adrian E. Raftery. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

[13] J. Hopcroft, O. Khan, B. Kulis, and B. Selman. Natural communities in large linked networks. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 541–546, 2003.

[14] D. Jensen, J. Neville, and B. Gallagher. Why collective inference improves relational classification. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 593–598, 2004.

[15] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, 1998.

[16] R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *ICML*, 2002.

[17] Y. Liu, R. Jin, and L. Yang. Semi-supervised multi-label learning by constrained non-negative matrix factorization. In *AAAI*, 2006.

[18] Q. Lu and L. Getoor. Link-based classification. In *ICML '03: Proceedings of the 20th international conference on Machine learning*, 2003.

[19] S. A. Macskassy and F. Provost. A simple relational classifier. In *Proceedings of the Multi-Relational Data Mining Workshop (MRDM) at the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[20] S. A. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *J. Mach. Learn. Res.*, 8:935–983, 2007.

[21] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–444, 2001.

[22] J. Neville and D. Jensen. Leveraging relational autocorrelation with latent group models. In *MRDM '05: Proceedings of the 4th international workshop on Multi-relational mining*, pages 49–55, 2005.

[23] J. Neville, D. Jensen, L. Friedland, and M. Hay. Learning relational probability trees. In *KDD*, pages 625–630, 2003.

[24] M. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 74(3), 2006.

[25] M. Newman. Modularity and community structure in networks. *PNAS*, 103(23):8577–8582, 2006.

[26] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[27] K. Nowicki and T. A. B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.

[28] P. Sarkar and A. W. Moore. Dynamic social network analysis using latent space models. *SIGKDD Explor. Newsl.*, 7(2):31–40, 2005.

[29] L. Tang, H. Liu, J. Zhang, N. Agarwal, and J. J. Salerno. Topic taxonomy adaptation for group profiling. *ACM Trans. Knowl. Discov. Data*, 1(4):1–28, 2008.

[30] L. Tang, H. Liu, J. Zhang, and Z. Nazeri. Community evolution in dynamic multi-mode networks. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 677–685, 2008.

[31] L. Tang, S. Rajan, and V. K. Narayanan. Large scale multi-label classification via MetaLabeler. In *Proceedings of the 18th international conference on World Wide Web*, 2009.

[32] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *UAI*, pages 485–492, 2002.

[33] J. Travers and S. Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):425–443, 1969.

[34] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

[35] K. Tsuda and W. S. Noble. Learning kernels from biological networks by maximizing entropy. *Bioinformatics*, 20:326–333, 2004.

[36] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[37] Z. Xu, V. Tresp, S. Yu, and K. Yu. Nonparametric relational learning for social network analysis. In *KDD'2008 Workshop on Social Network Mining and Analysis*, 2008.

[38] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.