2011 IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing

Extracting Social Dimensions using Fiedler Embedding

Xi Wang
Department of Electrical Engineering
and Computer Science
University of Central Florida
Orlando, Florida 32816
Email: xiwang@eecs.ucf.edu

Gita Sukthankar
Department of Electrical Engineering
and Computer Science
University of Central Florida
Orlando, Florida 32816
Email: gitars@eecs.ucf.edu

***Abstract*—In this paper, we present and evaluate the use of a Fiedler embedding representation for multi-label classification of social media. Networked data, such as data from social media, contains instances of multiple types that are related through different types of links. The network structure causes these data instances to no longer remain independently identically distributed (i.i.d.). Relational learning succeeds in improving the classification performance by leveraging the correlation of the labels between linked instances. However, instances in a network can be linked for different causal reasons, hence treating all links in a homogeneous way limits the performance of relational classifiers on such datasets. Social-dimension based approaches address this problem by extracting a feature space which captures the pattern of prominent interactions in the network. In this paper, we propose an alternate low-dimensional social feature representation that can be extracted from edge-based social dimensions using Fiedler embedding. This embedded feature space encodes the relations between people and their connections (nodes and links). Experiments on two real-world social media datasets demonstrate that our proposed framework offers a better feature representation for multi-label classification problems on social media.**

I. Introduction

Social media provides an enormous volume of information about individuals' personal information, opinions, and connections with others. This rich information enables the study of collective behavior within the social network—given a network structure, how to best predict one user's activities based on other individuals' behaviors within the same network. This problem has been referred to as the *node classification problem* [1], where users are represented as nodes in the network and the label associated with nodes denotes users' behaviors. In cases where behaviors are not mutually exclusive, this can be posed as a multi-label classification task. A number of relational classification models have demonstrated their advantages on collective classification [9], [10], [13]. However, one problem that often manifests is that different types of social connections are represented within the same set of links. In social networks, people usually connect with each other for different causal reasons. Although some tools such as Facebook contain mechanisms for grouping and tagging links, many social media datasets do not possess this additional information. Treating these inherently heterogeneous connections in a homogeneous way can result in erroneous classification results. Tang and Liu account for the possibility of connection heterogeneity by extracting social dimensions based on network connectivity [15], [16]. In this paper, we present an alternative method for constructing a low-dimensional social feature space by applying Fiedler embedding to an edge-based social feature space. Fiedler embedding [7] generates a geometric representation for nodes in a graph such that their geometric proximity approximates similarity information encoded within the graph edges. We demonstrate that, by capturing the correlations between different entities in the network (people and their connections), our embedded social feature space is capable of identifying semantically similar users within the social network and outperforms previous methods at identifying groups within two social media datasets.

II. Problem Formulation

People's actions in social networks are often highly influenced by their connectors—friends, relatives, colleagues and neighbors. Homophily, the phenomena that like-minded individuals have an increased propensity to be connected [12], can directly lead to behavior correlations between linked individuals. As a result of homophily, we expect to observe more links between people with the same affiliation than those with different affiliations.

However in social networks, interactions between people are commonly driven by multiple interests. Examples of interests include joining the same chat group, reading about the same topic, or watching the same online video. For this reason, social media links from a single person are generally not of homogeneous origin. If these connections are not grouped or tagged, it is possible to implicitly treat all of these connections as originating from the same type of social interaction. Link-centered representations ameliorate this problem by attempting to characterize the nature of the social connection separately from the affiliation.

In this paper, we aim to predict the collective behavior in the social media by utilizing behavior correlations embedded in the social network. Here, we treat this problem as a multi-label classification problem, since each person in the network can belong to one or more affiliations. Assuming that each

978-0-7695-4578-3/11 $26.00 © 2011 IEEE
DOI

person can be associated with K affiliations, a binary class vector, $C = \{C_1, C_2, ..., C_K\}$, can be used to represent the user's involvement in each affiliation. When $K = 1$ (each person only has a single affiliation), this problem is simplified to a binary classification problem.

We formulate our problem as follows: a society of N individuals are connected by the network graph $G = \{V, E, L\}$. In this graph, the set of nodes, $V = \{V_1, ..., V_N\}$, represent the social media contributors. The interactions between people are described by the edge set E. $L_i \subseteq L$ is the class label associated with node $V_i \in V$. The set of nodes, V, is further divided into two disjoint parts: X, the nodes for whom we know class labels (behavior categories), and Y, the nodes whose labels need to be determined. Our task is to determine the labels of the unknown nodes, Y, from the label set L, based on their interactions with other nodes (from X and Y) in the network.

III. Fiedler Embedding

Fiedler embedding was first proposed as a method for information retrieval and processing of document corpora [7]. It aims to map documents into a geometric space such that similar documents are close to each other. It has also been applied to the problem of video action recognition [8] to improve classification accuracy by combining two types of features. To the best of our knowledge, we are the first to apply Fiedler embedding to social network data. Here we use this technique to create a low-dimensional feature representation that encodes the relationship between nodes and edges implicitly contained within the edge cluster representation. The mathematical derivation of this embedding algorithm is as follows.

We begin with a graph $G = (V, E)$, where V is a set of vertices (nodes) and E is a set of edges represented by vertex pairs. An edge (V_i, V_j) connecting vertices V_i and V_j has a non-negative weight w_{ij} that describes the degree of similarity between two vertices. The more similar two vertices are to each other, the higher the corresponding weight. The vertices may represent several different classes of objects. For example, in the document case, the vertices are word terms and documents. Here, we assume that for any two vertices there is a connecting path.

The goal of Fiedler embedding is to project the vertices of the graph into a low-dimensional geometric space in such a way that similar vertices are close to each other even if they do not have a direct edge (observed relationship) between them. Following [7], this geometric embedding problem can be posed as a minimization problem. Specifically, we aim to find points in a k-dimensional space that minimize the weighted sum of squared edge lengths. If p_r and p_s are the locations of vertex r and s in the embedding space respectively, then our objective function can be written as follows:

$$\text{Minimize} \sum_{(r,s)\in E} W_{r,s}|p_r - p_s|^2 \tag{1}$$

Here, $W_{r,s}$ represents the weight of edge $E_{r,s}$. If the number of vertices is n, and the geometric space has dimensionality k, then the positions of the vertices can be considered to be an $n \times k$ matrix X. The Laplacian matrix L can be defined as follows:

$$L(i,j) = \begin{cases} -w_{i,j} & \text{if } e_{ij} \in E \\ \sum_k w_{i,k} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The Laplacian is the negative of the matrix of weights, except that the diagonal values are selected to make the row-sums zero. Thus L is symmetric and positive semi-definite. [7] reformulates the above minimization problem as:

$$\begin{aligned} &X = \arg\min_X \ \text{Trace}(X^T L X) \\ &\text{Subject to :} \\ &(i) X_i^T D 1^n = 0 \quad \text{for } i = 1, \ldots, k \\ &(ii) X^T D X = \Delta \end{aligned} \tag{3}$$

Here, two constraints are added. The first constraint is to make the median of the point set be the origin. 1^n denotes a vector of n ones. D is a positive diagonal weight matrix. The second constraint is to avoid the trivial solution of placing all the vertices at the origin. Δ denotes a non-negative diagonal matrix. The solution of this geometric optimization problem is $X = \tilde{Q}_k \Delta^{1/2}$, where $\tilde{Q}_k = [q_2, \ldots, q_{k+1}]$ are the generalized eigenvectors of $Ly = \lambda Dy$ sorted in non-decreasing order based on the corresponding eigenvalues λ_k. The solution is referred to as the Fiedler embedding.

In our case, two entities exist in the graph: social media users and edge cluster features. Fiedler embedding represents this bipartite graph in a low-dimensional space such that the geometric coordinates of similar nodes are closer to one other. From this point of view, Fiedler embedding is quite similar to the Co-Clustering method [2], which simultaneously clusters instances as features. However, the Fiedler embedding is more powerful in that it allows one to capture a set of relationship between different types of entities in the graph. An additional benefit is that the embedded features are easy to update if a new vertex is added to the graph. The position of the new vertex v in the embedded space can be calculated by:

$$p_v = \frac{\Delta^{1/2} \tilde{Q}_k^T q}{||q||_1} \tag{4}$$

where q is the vector of similarity values for the new vertex.

IV. Collective Behavior Classification Framework

In this section we describe our collective behavior classification framework using Fiedler embedding. In our work, we first extract the social features (SF) from the network topology using the edge clustering method described in [16]. These social features capture the nodes' involvement patterns in different potential affiliations. Once the features are computed, we apply the Fiedler embedding to discover the relationships

TABLE I
COLLECTIVE BEHAVIOR CLASSIFICATION FRAMEWORK

Input: Network data, the labels of training nodes
Output: Labels of the unlabeled nodes

1 Construct social feature space A using scalable edge K-means
- Convert the network into edge-centric view
- Perform k-means clustering on edges
- Create the social feature space based on the edge clustering using the *proportion* aggregation operator

2 Construct the Laplacian matrix L as:
- The user-feature similarity block is directly represented by the social feature space A.
- The user-user similarity block is computed using Equation 5, and the feature-feature similarity block is calculated as the inner product of the columns of A.

3 Perform eigen-decomposition on the Laplacian matrix L according to $L = VDV^T$. V and D are the eigenvectors and eigenvalues of L in non-descending order.

4 Construct the k-dimensional embedded space $S = D_k^{1/2}V_k^T$. D_k and V_k are the k smallest eigenvalues and corresponding eigenvectors omitting the first one whose eigenvalue equals zero.

5 Compute the embedded feature of the entity via mapping the entity to the k-dimensional embedded space: $D_k^{1/2}V_k^T q/||q||_1$.

6 Train the classifier based on the embedded social features of the labeled nodes.

7 Use the classifier to predict the labels of the unlabeled nodes based on their embedded feature.

among the nodes and edges by projecting them into a common Euclidean space. When applying Fiedler embedding, we first construct the Laplacian matrix from the whole dataset (both training and testing nodes), then the embedding space can be computed based on the k smallest eigenvalues (omitting the first one) of the Laplacian matrix. Finally, after mapping the social features into the embedding space, classification task can be performed on those embedded features. A detailed description of this collective behavior classification framework is provided in Table I.

A. Construct Laplacian Matrix

The important part of computing the Fiedler embedding is constructing the Laplacian matrix L, which is a symmetric matrix constructed according to Equation 2. In our case, we have two entities: the users and the social features (SF). The Laplacian matrix forms a a 2×2 block structure:

$$\begin{pmatrix} D1 & A \\ A^T & D2 \end{pmatrix}$$

Here $D1$, $D2$ and A denote the similarity matrix of user-user, SF-SF and user-SF respectively. In principle, the similarity relations between entities can employ a variety of suitable measurements such as inner product between features or co-occurrence.

Since the Fiedler embedding aims to find a k-dimensional space where the relationship between different entities is captured such that semantically similar nodes, even those that do not share links, are nearby in this embedded space. In our application, the Fiedler embedding provides a way to discover users with similar behaviors in the social network, even if they have not had direct interactions with one another. For this purpose, we must select a suitable method to measure the similarity between different entities. Depending on the type of entity, different similarity measurements may be needed. For instance, the user-user similarity can be evaluated by the inner product of the social features, whereas the probability of occurrence is a much better measure for user-SF similarity.

In our paper, since values in node's social features represent the node's probability of occurrence in the edge clusters (potential affiliations), we directly use the node's social feature extracted using *proportion* aggregation operator as the measure for the similarity between users and potential affiliations. The user-user similarity can be measured using functions such as cosine similarity, inner product or Gaussian kernel. Additionally, since users are connected with each other in the social network, network topology can also be a measure for the latent relationship between people in the network. In our case, it is reasonable to combine these two similarity measures to obtain a better results. If we denote W_f as the node affinity matrix obtained using social features and W_{adj} as the adjacency matrix of the social network, the combined similarity matrix can be obtained by the following schemes:

$$W_{\text{JOINT}} = W_f \circ W_{\text{adj}} \tag{5}$$

$$W_{\text{SUM}} = (1-\alpha)W_f + \alpha W_{\text{adj}} \tag{6}$$

The first equation calculates the affinity matrix in the joint space. $W_f \circ W_{\text{adj}}$ denotes a per-entry (Hadamard) product of W_f and W_{adj}. The second equation defines the affinity matrix by adding W_f and W_{adj} together (α is a number between 0 and 1). The choice of the similarity function depends mainly on the data. Through our pilot experiments, we determined that W_{JOINT} performs better at clustering similar users and select as in our default user-user similarity measure. Since it is desirable to use the same measure to calculate SF-SF similarity, the method should be able to capture the relationship for both user-user and SF-SF. In our experiment, we observe that both the cosine distance and the Gaussian kernel can group the similar people together, but they were unable to group similar social features. However, the inner product method performs well at grouping both people and social features. Thus, we choose the inner product to measure the similarity for user-user and SF-SF.

V. EXPERIMENTAL SETUP

A. Social Media Datasets

To facilitate direct comparison, we follow the evaluation protocol of [15]. Our proposed method is evaluated on real social network datasets extracted from two popular social media tools, BlogCatalog and Youtube. Both datasets are available from the Data Mining and Machine Learning Lab at http://www.public.asu.edu/~ltang9/social_dimension.html.

BlogCatalog: A blog in BlogCatalog can be associated with various pieces of information such as the categories the blog is listed under (e.g., "Music", "Education" and "Sports"), the blog subcategory tags (e.g., "Pop", "Science" and "Football") and the blog post level tags (e.g., "pop singers", "biology" and

TABLE II
DATA STATISTICS

Data	BlogCatalog	YouTube
Categories	39	47
# of Nodes	10312	15000
# of Links	333,983	136,218
Network Density	6.3×10^{-3}	1.2×10^{-3}
Maximum Degree	3,992	14,999
Average Degree	65	9
Average Categories	1.6	2.1

"top team"). Each blog category can be treated as a label that denotes the blogger's personal interest. Moreover, the blogger can specify his/her social connections with other bloggers. The blogCatalog dataset contains 39 categories and the average number of categories that each instance (blog) belongs to is 1.6.

YouTube is a popular website for sharing videos. Each user in Youtube can subscribe to different interest groups and add other users as his contacts. In this paper, we select a small subset of data (15000 nodes) from the original Youtube dataset in [16] using *snowball sampling*, and retain 47 interest groups as our class label. The details of the data set can be found in Table II.

B. Baseline Methods

In this paper, we compare our proposed embedded social feature extraction method to five related methods: *EdgeCluster*, *wvRN*, *NodeCluster*, *Co-Clustering* and *Random*. A short description of these methods follows:

• *EdgeCluster* denotes the best performing method described in [16], where the social dimensions are directly extracted using the edge clustering representation. The edge-based social features are constructed using the *proportion* operator, and a linear SVM is used for discriminant learning. We show the advantage of our embedded features by exploiting the relationship between different edge clusters.

• Weighted-vote Relational Neighbor Classifier (*wvRN*) [11] is a simple relational model that makes predictions based solely on the class labels of the related neighbors. The node's predicted class memberships are regarded as the weighted mean of the its neighbors. Though *wvRN* is a classifier which doesn't require learning, it performs surprisingly well on networked datasets.

• *NodeCluster* is based on [14] which assumes that each node is associated with only one affiliation.. In this paper, we adopt the edge-based social dimensions as our raw social features because it allows one node to possess multiple affiliations. To verify this concept, we compare *EdgeCluster* with the *NodeCluster* method. For comparison, we first adopt k-means clustering to partition the network into disjoint groups, and then construct the social features using node clustering IDs as features. This comparison scheme is also examined in [16].

• *Co-Clustering* is a clustering method for partitioning the instances and their features simultaneously [2]. It was first proposed for clustering the document datasets represented as a bipartite graph between documents and their words. The k-means algorithm is applied on the top singular vectors of the scaled document-word matrix (omitting the principal singular vector). The main difference between Co-clustering and Fiedler embedding is that the Co-clustering algorithm does not take into account the similarity between different entities during its clustering procedure.

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

In our proposed method, Fiedler embedding is used to extracted the embedded social features. First, the edge clustering method is adopted to construct the initial social dimensions. We use cosine similarity while performing the clustering; the dimensionality of the edge-based social features is set to 1000 for both BlogCatalog and Youtube dataset. Then, the embedded social features are extracted from the raw social dimensions using the procedure described in [16]. Finally, a set of one-vs-all support vector machines (SVMs) are employed for classification.

Since our classification problem is essentially a multi-label task, during the prediction procedure, we assume that the number of labels for the unlabeled nodes is already known and assign the labels according to the top-ranking class. Such a scheme has been adopted for multi-label evaluation in social network datasets [15], [16]. In our work, we sample a small portion of nodes uniformly from the network as training instances. The fraction of the training data is from 5% to 30% for BlogCatalog dataset, and 1% to 10% for the Youtube datast. Two commonly used measures Micro-F1 and Macro-F1 are adopted to evaluate the classification performance [3].

VI. RESULTS

To evaluate the performance of the Fiedler embedding representation, we compare the classification results of our proposed framework against five baseline methods in BlogCatalog and Youtube dataset. Tables III and IV list the classification performance of all methods in the BlogCatalog dataset and Youtube dataset, respectively. The best classification rates under each training condition are shown in bold. As we can see from the table, the methods based on edge-based social features; *EdgeCluster*, *EdgeCluster+Co-Clustering* and *EdgeCluster+Fiedler*, generally outperform the other non-edge based methods. The baseline method *Random* only achieves around 4% for Macro-F1 while the other methods reach above 10% on the BlogCatalog dataset. *wvRN* method clearly outperforms *Random* method by utilizing correlation between linked nodes in the social networks for prediction. However, without differentiating the connections, *wvRN* shows poor performance when the links in network are heterogeneous. This is especially noticeable when we compare the results in Youtube dataset. By assuming each user is involved in one affiliation, the *NodeCluster* method performs worse than *EdgeCluster*.

Clearly, our proposed method *EdgeCluster+Fiedler* consistently outperforms *EdgeCluster* method in both BlogCatalog and Youtube datasets. By exploiting the correlations between different potential affiliations (edge clusters), Fiedler embedding shows its advantages in identifying the similar instances especially when the amount of training data is small. In the Youtube dataset, social features based on Fiedler embedding improves the classification results of *EdgeCluster* by 7% for Micro-F1 when only 1% of the instances are sampled as training data. Moreover, the higher results for Macro-F1 also demonstrate that the embedded social features has better performance in distinguishing different types of connections among the network. Fiedler embedding boosted the Macro-F1 results by around 2% and 5% in BlogCatalog and Youtube datasets respectively. Unfortunately, the comparison method, Co-Clustering, which is not able to capture the correlations between nodes and edge clusters, performs poorly in grouping multi-label instances and even undermines the performance of *EdgeCluster*.

VII. Related Work

The area of Statistical Relational Learning (SRL) [4] focuses on learning models of networked data—objects or entities that are represented by an uncertain, complex and relational structure. In this work, we study a special case in SRL, within-network classification [10], when all the objects are connected in one network. The network structure makes the data instances no longer independently identically distributed (i.i.d.). Relational classifiers have the ability to improve the performance over traditional classifiers by taking advantage of the dependencies of both labels, and sometimes attributes, of related labeled instances [10].

Collective inference based on network connectivity is often adopted for prediction. The weighted-vote relational neighbor (wvRN) classifier [11] is a simple relational classifier that predicts the label of the unknown node by taking the weighted average of the potential estimated class membership scores of its neighbors. Neville et al. proposed a simple iterative classification algorithm to classify correlated entities [13]. Specifically, a relational classifier is trained using a combination of entities' static features and relational features. The relational feature is calculated by aggregating the *inferred* intrinsic attributes of related objects. [9] extended this simple classifier by introducing a relational feature vector that measures the distribution of the class labels among directly linked objects.

Collective inference in relational learning relies on the assumption that the connections in the network are homogeneous. However, different types of relationships commonly coexist in social network. Goldberg et al. make the observation that in social media, nodes may link to one another even if they do not have similar labels [5]. In the paper, they use two edge types to denote the affinity or disagreement in the class labels of linked objects and incorporates the link type information into discriminant learning. [6] proposed a Link Type Relational Bayes Classifier that predicts the node's class labels according to the neighbors' labels as well as their link types. Tang et al. proposed the *SocDim* framework to directly address the link heterogeneity problem. Latent social dimensions are extracted from the network using modularity maximization to capture the potential affiliations of the entity, and then a discriminant classifier is adopted for prediction based on the social dimensions. [17] uses an alternate spectral clustering approach to to extract social dimensions. Both of these approaches extract social dimensions from a node-based network representation and achieve superior performances than the commonly used relational classifiers, wvRN and LBC. Our embedded social feature space is extracted from the edge clustering representation proposed by [16], which has shown comparable results to [15]. However, in our work, instead of treating the node and edge-cluster as independent entities, we adopt Fiedler embedding to find a low-dimensional feature space that encodes their relations. The Fiedler embedding can recover from inefficiencies in the edge clustering representation by discovering similarities between clusters.

Note that Fiedler embedding is quite related to several popular techniques in information retrieval. One is Latent Semantic Analysis (LSA) [18], which aims to find a geometric space based on a document-term matrix of the document collection. When calculating the mapping space, LSA is concerned with documents and terms while Fiedler embedding concentrates on general entities and their similarities. Unlike LSA, Fiedler embedding treats all entities (i.e., document and term) equivalently and as co-located in the same space. From this point of view, Fiedler embedding is also quite similar to the Co-Clustering method [2], which places the instances (document) and their features (terms) in the same space and performs clustering on both entities simultaneously. In this paper, we adopt Co-Clustering as one of our comparison methods and evaluate its performance on edge-based social features.

VIII. Conclusion

The heterogeneity of connection types in social media datasets creates issues when attempting to predict users' behaviors based on its neighbors. To solve this problem, in this paper we exploit the geometric properties of Fiedler embedding to construct an alternate social feature space from edge-based social features. Comparative experiments on two social media datasets demonstrate that this embedded social feature space possesses the following advantages: 1) it encodes the similarity between different entities (users and their connections), 2) is a better representation for distinguishing different types of connections and predicting collective behavior, 3) is able to discover semantically similar users who are disconnected in the network. In future work, we plan to explore the performance of Fiedler embedding at link prediction between similar users in social network datasets. By identifying users who are similar in the Fiedler embedding space and recommending them as possible connections, we can enhance the user experience of various social media tools.

TABLE III
CLASSIFICATION PERFORMANCE FOR BLOGCATALOG DATASET

Proportion of Labeled Nodes		5%	10%	15%	20%	25%	30%
Micro-F1(%)	*EdgeCLuster+Fiedler*	**23.72**	**24.25**	**26.15**	**26.85**	**27.43**	**27.45**
	EdgeCluster+Co-Clustering	18.55	19.92	21.30	21.94	22.34	22.67
	EdgeCluster	19.44	22.72	24.92	26.02	26.98	27.35
	wvRN	14.28	17.80	21.04	22.55	24.66	25.42
	NodeCluster	10.04	17.25	18.29	19.00	19.51	19.79
	Random	4.84	4.67	4.76	4.71	4.67	4.76
Macro-F1(%)	*EdgeCLuster+Fiedler*	**14.12**	**16.05**	**16.97**	**17.90**	**18.11**	**18.87**
	EdgeCluster+Co-Clustering	10.90	11.86	12.60	13.25	13.52	13.96
	EdgeCluster	12.12	14.63	15.89	17.30	17.75	18.55
	wvRN	8.59	10.54	12.29	12.99	14.20	14.48
	NodeCluster	9.47	11.08	12.06	12.74	13.31	13.66
	Random	4.07	3.97	4.04	4.01	3.97	4.07

TABLE IV
CLASSIFICATION PERFORMANCE FOR YOUTUBE DATASET

Proportion of Labeled Nodes		1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
Micro-F1(%)	*EdgeCLuster+Fiedler*	**34.11**	**34.80**	**35.58**	**36.28**	**37.08**	**37.94**	**38.02**	**38.10**	**38.16**	**38.21**
	EdgeCluster+Co-Clustering	23.00	25.40	26.86	27.60	28.92	29.09	29.11	30.53	31.19	31.52
	EdgeCluster	27.52	27.97	28.12	28.91	29.94	31.01	31.28	31.34	32.95	32.98
	wvRN	13.61	15.10	16.21	17.04	17.79	18.72	19.31	19.87	20.44	21.04
	NodeCluster	24.44	24.12	23.77	24.48	25.11	24.90	25.29	25.57	26.06	26.57
	Random	9.36	9.64	9.70	9.94	9.96	9.88	9.71	9.77	9.99	9.76
Macro-F1(%)	*EdgeCLuster+Fiedler*	**22.13**	**23.73**	**25.43**	**26.38**	**27.07**	**28.37**	**28.71**	**29.18**	**29.60**	**29.91**
	EdgeCluster+Co-Clustering	16.22	18.26	19.68	20.50	21.40	22.13	22.56	23.22	23.85	24.31
	EdgeCluster	17.22	19.10	20.45	21.33	22.30	23.29	23.50	23.93	24.93	25.03
	wvRN	11.38	13.10	14.51	15.41	16.34	17.41	17.99	18.75	19.18	19.66
	NodeCluster	19.52	20.15	20.93	21.65	22.38	23.17	23.32	23.78	24.10	24.49
	Random	8.69	8.98	9.03	9.27	9.18	9.32	9.13	9.14	9.29	9.08

ACKNOWLEDGMENTS

We thank Huan Liu and Lei Tang for making their datasets publicly available. This research was supported in part by NSF award IIS-0845159.

REFERENCES

[1] S. Bhagat, G. Cormode, and S. Muthukrishnan. Node classification in social networks. *Computing Research Repository (CoRR)*, abs/1101.3291, 2011.

[2] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 01)*, pages 269–274, 2001.

[3] R.-E. Fan and C.-J. Lin. A study on threshold selection for multi-label classication. Technical report, National Taiwan University, 2007.

[4] L. Getoor and B. Taskar. *Introduction to statistical reltional learning*. The MIT Press, 2007.

[5] A. Goldberg, X. Zhu, and S. Wright. Dissimilarity in graph-based semi-supervised classification. In *Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS 07)*, 2007.

[6] R. Heatherly, M. Kantarcioglu, and X. Li. Social network classification incorporating link type. In *Proceedings of IEEE Intelligence and Security Informatics (ISI 09)*, pages 19–24, 2009.

[7] B. Hendrickson. Latent semantic analysis and Fiedler retrieval. *Linear Algebra and Its Applications*, 421:345–355, 2007.

[8] J. Liu, S. Ali, and M. Shah. Recognizing human actions using multiple features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 08))*, pages 1–8, 2008.

[9] Q. Lu and L. Getoor. Link-based classification. In *Proceedings of 20th International Conference on Machine Learning (ICML03)*, pages 496–503, 2003.

[10] S. Macskassy and F. Provost. Classification in networked data: a toolkit and a univariate case study. *Journal of Macnine Learning*, 8:935–983, 2007.

[11] S. A. Macskassy and F. Provost. A simple relational classifier. In *Proceedings of the Second Workshop on Multi-Relational Data Mining (MRDM 03) at KDD 2003*, pages 64–76, 2003.

[12] M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

[13] J. Neville and D. Jensen. Iterative classification in relational data. In *Proceedings of the AAAI 2000 Workshop Learning Statistical Models from Relational Data*, pages 42–49, 2000.

[14] J. Neville and D. Jensen. Leveraging relational autocorrelation with latent group models. In *Proceedings of the 4th International Workshop on Multi-relational Mining (MRDM 05)*, pages 49–55, 2005.

[15] L. Tang and H. Liu. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining (KDD 09)*, pages 817–826, 2009.

[16] L. Tang and H. Liu. Scalable learning of collective behavior based on sparse social dimensions. In *Proceedings of International Conference on Information and Knowledge Management (CIKM 09)*, pages 1107–1116, 2009.

[17] L. Tang and H. Liu. Leveraging social media networks for classification. *Data Mining and Knowledge Discovery (DMKD 2011)*, 23, 2011.

[18] D. L. Thomas K. Landauer, Peter W. Foltz. An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284, 1998.