Node Classification in Social Network via a Factor Graph Model

Huan Xu[1], Yujiu Yang[1], Liangwei Wang[2], and Wenhuang Liu[1]

[1] Shenzhen Key Laboratory of Information Science and Technology,
Graduate School at Shenzhen, Tsinghua University, P.R. China
[2] Noah's Ark Laboratory, Huawei Technologies Co., Ltd., Shenzhen, P.R. China

Abstract. This paper attempts to addresses the task of node classification in social networks. As we know, node classification in social networks is an important challenge for understanding the underlying graph with the linkage structure and node features. Compared with the traditional classification problem, we should not only use the node features, but also consider about the relationship between nodes. Besides, it is difficult to cost much time and energy to label every node in the large social networks. In this work, we use a factor graph model with partially-labeled data to solve these problems. Our experiments on two data sets (DBLP co-author network, Weibo) with multiple small tasks have demonstrated that our model works much better than the traditional classification algorithms.

1 Introduction

With the success of many large-scale online social networks these years, such as Facebook, Tweeter, Weibo, social network has become a bridge between the virtual web world and our daily life. Weibo, one of the largest and most influential social networks in China, has more than 300 million active users in 2012. Consequently, Network ideas have been applied successfully in many areas such as Internet (pages) [3][8], coauthors[19], mobiles and mails[5]. Considerable research has been conducted on social network analysis, such as social influence analysis[6][18], community structure learning[18][1][14], and of course node classification in social network[4][15][7][10].

As is usual in machine learning, we first have to identify some features of nodes that can be used to guide the classification. The obvious features are properties of the node itself. For online social network like Weibo, information that may be known for all (or most) nodes, such as age, location, and some other profile information is usually considered first. For coauthor network, people may take authors profile, publish year as features. More than that, some latent features such as topic model become more and more popular in network analysis, these latent features reflect user character quite well. But in a network structure, the presence of an explicit link structure makes the node classification problem different from traditional machine learning classification tasks, where objects being classified are considered independent. In contrast to the traditional classification,

J. Pei et al. (Eds.): PAKDD 2013, Part I, LNAI 7818, pp. 213–224, 2013.

we should first think about the network topology, the neighbors label information may be very important and decisive. The social sciences identify two important phenomena that can apply in social networks:

- **homophily**, when a link between individuals (such as friendship or other social connection) is correlated with those individuals being similar in nature. For example, friends usually have the similar age, education background.
- **co-citation**, regularity is a related concept, which holds when similar individuals tend to refer or connect to the same things. For example, when two people send microblogs with similar topics, it will be probably they have similar taste in some areas.

Macskassy and Provost used a simpler classification method based on taking a weighted average of the class probabilities in the neighborhood (wvRN)[11]. This classifier is based on a direct application of homophily and uses the immediate neighborhood of a node for classification. Another classifier, which is similar with wvRN, is called the Class-Distribution Relational Neighbor (CDRN)[12], it also considered on the distribution of the neighbors only. There were many works about using random walk on node classification, but most of them concerned about the labels of neighbors. Pennacchiotti and Popescu[15] used a machine learning algorithm with hundreds of features in twitter data, but they simply take network topology as some features in a traditional classifier.

In this work we address the task of node classification in social networks, our main contributions are the following:

- We employ a factor graph model to classify nodes in networks, using topic model as node features instead of profiles and consider multiple relationships in network.
- We conduct experiments on two data sets (DBLP coauthor and Weibo) and multiple tasks. Experimental results show that our model can be applied to the different scenarios and perform quite well than traditional classifiers.
- We provide an in-depth analysis of experiments on the partially-labeled data sets, results show that our model can do a good job with different ratio of labeled data.

The rest of this paper is organized as follows. Section 2 formally formulates the problem. Section 3 explains the factor graph model we used. And then in section 4 we discuss the experiments and evaluation. Finally, in Section 5 we draw final conclusions and outline future work.

2 Problem Definition

In this section, we first give several necessary definitions and then present the problem formulation.

Definition 1. ***Social network:*** *A social network can be represented as $G = (V, E)$, where V is a set of $|V| = N$ users. $E \subset V \times V$ is a set of $|E| = M$ relationships between N users.*

As we know, Social relationships might be directed in some networks (e.g., A follow B in Weibo) or undirected in others (A and B are coauthors in DBLP). In this work, we make the relationship as factor and ignore whether it is directed or undirected, the number of relationship types is what we only concern about. In some situations, the network can be defined in more than one way, such as in DBLP coauthor network: take each author as a node, coauthor may be a relationship; we can also take each paper as a node, two papers have common author may be a relationship. Which one should we choose is depend on the specific task, in this work, we choose the latter.

In real networks, it is difficult to cost much time and energy to label every node, the labeled data is always less than expected, so naturally, we define the input of our problem, a partially labeled network.

Definition 2. ***Partially labeled network:*** *A partially labeled network is an augmented social network represented as $G = (V_L, V_U, E, Y_L, W)$, where V_L is a set of labeled nodes while V_U is a set of unlabeled nodes with $V_L \bigcup V_U = V$; Y_L is a set of labels corresponding to the node classes in V_L; W is an weight matrix associated with users in V where each row corresponds to a user, each column an attribute, and an element w_{ij} denotes the value of the j^{th} attribute of user v_i.*

In our work, we choose a topic model called PLSA (Probabilistic Latent Semantic Analysis)[9] to analysis the text about each node as attribute. The PLSA model assumes that there are k topics in the corpora, where k is a fixed parameter, and every document in the corpora corresponds to one distribution of topics. This is a hierarchical model. We can describe its generative process as:

- *Select a document d with probability $P(d)$;*
- *Pick a latent topic z with probability $P(z|d)$;*
- *Generate a word w with probability $P(w|z)$.*

For example, we use PLSA to analysis the microblogs of every user in Weibo, and w_{ij} represents the probability of user v_i belongs to j^{th} topic.

Based on the above concepts, we can now define the problem of node classification in social network. Given a partially labeled network, the goal is to detect the classes (labels) of all unknown nodes in the network. Formally,

Target 1. ***Node classification in social network:*** *Given a partially labeled network $G = (V_L, V_U, E, Y_L, W)$, the objective is to learn a predictive function*

$$f : G = (V_L, V_U, E, Y_L, W) \rightarrow Y$$

The above formulation make our work very different from existing work on node classification. Macskassy, C. Perlich and Desrosiers[4][11][12] had done some constructive work about relational learning algorithm, but they only concern about the relationships in network. Pennacchiotti[15] tried many features about each node, even treated relationship as some features, but he ignored the network topology.

3 A Factor Graph Model

Actually in social network analysis, there are some works about utilizing graph model. For example, Tang[17] used a factor graph model to infer social ties; Yang[18] proposed a factor graph model too, their works focused on representative-user finding and community-structure discovery. In this section, we explain the factor graph model we use to classify nodes in social network.

3.1 Basic Domain Criterions

For inferring label of nodes in social network, we have three basic criterions from our specified domain knowledge.

First, users from different classes may have different topics while in same class may have similar topics. For example, if two users come from the same company, they may have similar microblogs in Weibo which talk about their works or about their company. Second, the relationships may have a correlation in classification. For example, in DBLP network, if two papers have at least one common author, they probably belong to the same research area (e.g., artificial intelligence, database and so on).

And last, different relationship types may lead to different kinds of effects. For example in Weibo, user A follows user B but B doesnt follow A back, this is a relationship type; user A follows user C and user B also follows C, this is another relationship type between A and B, obviously, these two relationship types are quite different, and of course it will influence the classification model.

3.2 A Factor Graph Model with Partially-Labeled Data

Based on the above observations, we then describe the proposed factor graph model in details.

As we can see in figure 1, it is a small network with 4 nodes v_1, v_2, v_3, v_4 and some relationships between these nodes. Corresponding to the factor graph model, we have 4 hidden vectors y_1, y_2, y_3, y_4 and the nodes attribute factor. To deal with multiple relationship types, we should have multiple relation factors. Actually, we use at most 2 relation factors in our experiments, but in theory, we can use as many relation factors as possible. The definitions of the factors are as follows:

- ***Attribute factor:*** *$f(y_i, w_i)$ represents the posterior probability of the relationship y_i given the attribute vector w_i;*
- ***Relation factor 1:*** *$g(y_i, y_j)$ denotes a relationship of node y_i and y_j;*
- ***Relation factor 2:*** *$h(y_i, y_j)$ denotes another relationship of node y_i and y_j.*

Given a partially-labeled network $G = (V_L, V_U, E, Y_L, W)$, based on the definition of factor graph model, we can have the joint distribution over Y as

$$p(Y|G) = \prod_i f(y_i, w_i) g(y_i, N_1(y_i)) h(y_i, N_2(y_i)) \tag{1}$$

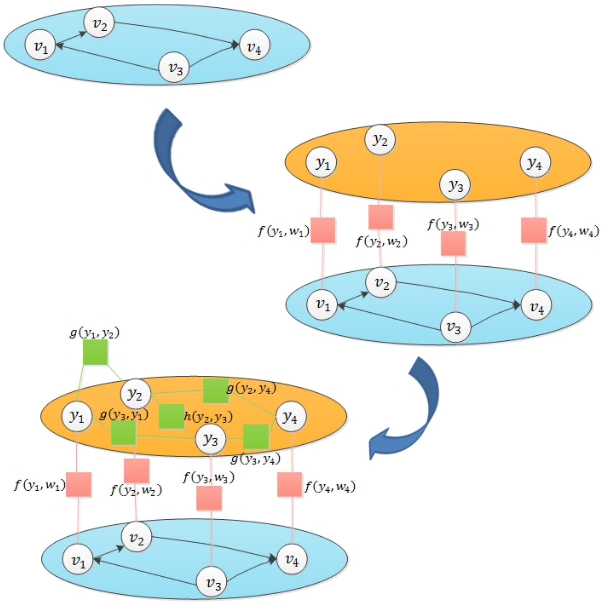

Fig. 1. A Factor Graph Model

where $N_1(y_i)$ and $N_2(y_i)$ are sets of neighbors of y_i. The three factors can be instantiated in different ways. In this work, we use exponential-linear functions. Particularly, we define the attribute factor as

$$f(y_i, w_i) = \frac{1}{Z_\lambda} exp\{\lambda^T \varPhi(y_i, w_i)\} \tag{2}$$

where λ is a weighting vector that will be learned in the model and $\varPhi$ is a vector of feature functions. Similarly, we denote the relation factor as

$$g(y_i, N_1(y_i)) = \frac{1}{Z_\alpha} exp\{\sum_{y_i \in N_1(y_i)} \alpha^T g(y_i, y_j)\} \tag{3}$$

$$h(y_i, N_2(y_i)) = \frac{1}{Z_\beta} exp\{\sum_{y_i \in N_2(y_i)} \beta^T h(y_i, y_j)\} \tag{4}$$

where α and β is similar with λ, g and h can be defined as a vector of indicator functions.

3.3 Model Learning and Parameter Inference

Learning this model is to estimate a series of parameters $\theta = (\lambda, \alpha, \beta)$ and maximize the log-likelihood of observation information (labeled nodes). For simple presentation, we concatenate all factor functions for y_i as

$$s(y_i) = (\varPhi(y_i, w_i)^T, \sum_{y_j} g(y_i, y_j)^T, \sum_{y_j} h(y_i, y_j)^T)^T \tag{5}$$

The joint probability can be written simply as

$$\begin{aligned} p(Y|G) &= \frac{1}{Z}\prod_i exp\{\theta^T s(y_i)\} \\ &= \frac{1}{Z}exp\{\sum_i \theta^T s(y_i)\} \\ &= \frac{1}{Z}exp\{\theta^T S\} \end{aligned} \tag{6}$$

where $Z = Z_\lambda Z_\alpha Z_\beta$ is a normalization factor, and S is the combination of factor functions of all nodes.

Since the input data of this model is partially-labeled, to calculate the normalization factor Z, we need to sum up the likelihood of possible states for all labeled and unlabeled nodes. Naturally, we think of using the labeled data to infer the label of unknown nodes. Then, we have the objective function as

$$\begin{aligned} O(\theta) &= log\sum_{Y|Y^L} \frac{1}{Z}exp\{\theta^T S\} \\ &= log\sum_{Y|Y^L} exp\{\theta^T S\} - logZ \\ &= log\sum_{Y|Y^L} exp\{\theta^T S\} - log\sum_{Y} exp\{\theta^T S\} \end{aligned} \tag{7}$$

Where $Y|Y^L$ denotes inferring label of Y from Y^L. To solve this problem, we use a gradient decent method to calculate the partial derivative of θ

$$\begin{aligned} \frac{\partial O(\theta)}{\partial \theta} &= \frac{\partial(log\sum_{Y|Y^L} exp\{\theta^T S\} - log\sum_Y exp\{\theta^T S\})}{\partial \theta} \\ &= \frac{\sum_{Y|Y^L} exp\{\theta^T S \cdot S}{\sum_{Y|Y^L} exp\{\theta^T S} - \frac{\sum_Y exp\{\theta^T S \cdot S}{\sum_Y exp\{\theta^T S} \\ &= E_{\theta(Y|Y^L,G)}S - E_{\theta(Y,G)}S \end{aligned} \tag{8}$$

Since the social network can be arbitrary graphical structure, our factor graph model may have many circles, so it is intractable to calculate the expectation in a directed and exact way. To alleviate the cost of computation, some kinds of approximate algorithms have been proposed such as LBP (Loopy Belief Propagation)[13] and MCMC (Markov Chain Monte Carlo)[16]. In this work, we utilize LBP to calculate the marginal probabilities.

LBP is simply to apply the sum-product algorithm even though there is no guarantee that it will yield good results[2]. It is possible because the message passing rules for the sum-product algorithm are purely local. However, because the graph now has cycles, information can flow many times around the graph. For some models, the algorithm will converge, whereas for others it will not.

After completing the calculation of the marginal probabilities, the gradient can be obtained by summing over all nodes, and then we update each parameter with a learning rate η and the gradient.

3.4 Infer Unknown-Label Nodes and Classify

Finally, we can infer the category of unknown-label node. Based on the learned parameters θ, we can predict the label of each node by finding a label configuration which maximizes the joint probability as

$$\tilde{Y} = argmax_{Y|Y^L} p(Y|G) \tag{9}$$

In the same way, we use LBP to calculate the marginal probability of each node $p(y_i|Y^L, G)$, then we can put one node into the class which has the maximum marginal probability. In other words, the marginal probability is taken as the prediction confidence.

4 Experiment Results

The model we use to classify nodes is general and can be used in many different situations. In this section, we present our experiment results on two different data sets with multiple tasks to evaluate the effectiveness of our model.

4.1 Data Sets

DBLP Coauthor Network. This benchmark data set contains more than 50000 papers published at 22 computer science conferences from 2008 to 2010. These conferences can be mainly divided into five research areas:

- AI: artificial intelligence, including IJCAI, AAAI, ICML, UAI and NIPS;
- DB: database, including EDBT, ICDT, ICDE, PODS, SIGMOD and VLDB;
- DP: distributed and parallel computing, including ICCP, IPDPS and PACT;
- GV: graphics, vision and HCI, including ICCV, CVPR and SIGGRAPH;
- NC: networks, communications and performance, including MOBICOM, INFOCOM, SIGMETRICS and SIGCOMM.

In this data set, the objective is to classify papers into the correct research area. We extract some subsets randomly for three tasks:

- 6000 papers with 3000 papers published in GV (positive set) and 3000 in others (negative set), 25319 edges totally;
- 6000 papers with 3000 papers published in NC (positive set) and 3000 in others (negative set), 18085 edges totally;
- 10000 papers with 5000 papers published in AI (positive set) and 5000 in others (negative set), 48083 edges totally.

Weibo Network. As our previous introduction, Weibo is a very large online social network with 300 million users. We extract some small experiment data from Weibo by crawler. In Weibo data, we have two binary classification tasks as:

- **Company Affiliation:** 600 users with 305 belong to a specific company (positive set) and 295 are not (negative set), 2393 relationships of following others, 2509 relationships of following common users. The positive users explicitly mention their company in tags, descriptions or screen names;

- **Domain Affiliation:** 3231 users with 1580 positive users whose study domain is about internet information technique and other 1651 users are negative with some different domains, 12194 relationships of following others, 2740 of following common users. The users we collect are explicitly mention their domain in tags.

Table 1. statistics of the data sets

Data set	Task	Users	Positive set	Negative set	Relationship (type1+type2)
DBLP	GV	6000	3000	3000	25319
	NC	6000	3000	3000	18085
	AI	10000	5000	5000	48083
Weibo	company	600	305	295	2393+2509
	domain	3231	1580	1651	12194+2740

4.2 Experiment Description and Evaluation

In the DBLP data set, we make each paper as a node, if two papers have at least one common author, we treat this relationship as an edge. In DBLP data, we simply use one relationship type. In Weibo data set, we treat each user as a node, if a user follows another user, we put an edge between them. More in Weibo data, we consider about the relationship of following common users (e.g. two users follow at least 5 common users) and make it as another edge factor.

In both two data sets, we use PLSA to get the topic model of each node as the node features. Actually, we set the number of topics as 20.

To compare our approach with the traditional methods, we carried out the representative algorithms such as wvRN, CDRN, LibSVM on the same data sets:

- **wvRN:** a simply neighbor-voted classifier, consider about the relationship only.
- **CDRN:** similar with wvRN, it uses the probability distribution of neighbors to classify nodes, consider about the relationship only.
- **libSVM:** a traditional classification algorithm, using node features to classify nodes. We use Weka 3.6 to implement libSVM.

To quantitatively evaluate the model we use, we consider three aspects:

- **Results Analysis (Basic Experiment):** we try to prove the effectiveness of our method with some basic experiments.
- **The Influence of Labeled Data Size:** since our method is semi-supervised, we use the different labeled data size to test the robustness and generality of our method.
- **Discussion on Multiple Relationship Types Setting:** we consider about multiple relationship types in our experiments and discover the extendibility of our method.

For the classification performance, we evaluate the approaches in terms of accuracy, precision, recall, and F1-score.

4.3 Accuracy Performance

Basic Experiment. Table 2 lists the accuracy performance of classifying nodes in DBLP data with three tasks by the different methods. All these three tasks, the labeled data is 10% of each subset.

Table 2. performance of node classification with different methods on three tasks in DBLP data

Positive set	Method	Accuracy	Precision	Recall	F1-score
GV	wvRN	0.580	**0.949**	0.469	0.627
	CDRN	0.584	0.935	0.479	0.633
	libSVM	0.860	0.818	0.925	0.868
	Our model	**0.943**	0.921	**0.962**	**0.944**
NC	wvRN	0.522	**0.984**	0.365	0.532
	CDRN	0.523	**0.984**	0.365	0.532
	libSVM	0.754	0.679	0.963	0.796
	Our model	**0.913**	0.891	**0.942**	**0.916**
AI	wvRN	0.448	**0.961**	0.272	0.424
	CDRN	0.451	**0.961**	0.273	0.426
	libSVM	0.834	0.814	0.868	0.840
	Our model	**0.897**	0.889	**0.907**	**0.898**

As we can see from the results, our method consistently outperforms other comparative methods on all the three tasks in DBLP data. In terms of F1-score, our model is 5% – 12% better than libSVM, 30% – 40% better than wvRN and CDRN. We notice that wvRN and CDRN, which are focus on the labels of neighbors, get very high precision scores, even better than our model, but their recall scores are really poor, and their F1-score are much worse than libSVM and our model, so wvRN and CDRN dont classify the nodes well. Actually, these two methods classify most of the nodes into negative set.

Different Size of Labeled Data. We have known that in DBLP data, our model perform much better than others and either wvRN or CDRN has a poor result. One of the reasons maybe the labeled data is only 10%. Based on the above assumption, we have some experiments on company affiliation using Weibo data, the classification objective is estimate whether a user is from a specific company or not. We test the size of labeled data from 10% to 90%, and the results are shown in figure 2.

From the results shown in figure 2, we can find some interesting points. When labeled data are 10%, our model is 11.9% better than libSVM, about 30% better than wvRN and CDRN in F1-score; on accuracy score, our model is also 10% better than others. When labeled data are 50%, on F1-score, our model is still 11.8% better than libSVM, and about 35% better than wvRN and CDRN; on accuracy score, our model is more than 20% than others. When labeled data

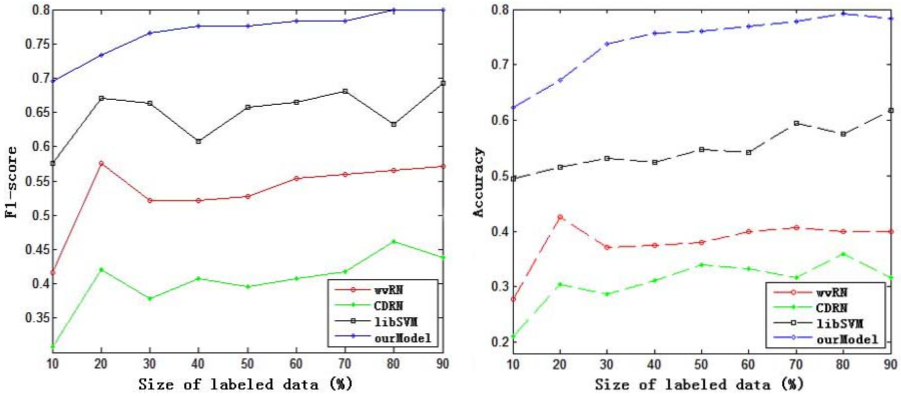

Fig. 2. Performance of node classification with different labeled size and different method on company affiliation using Weibo data

are 90%, our model is 11.7% better than libSVM and more than 25% better than wvRN and CDRN on F1-score; on accuracy score, our model is more than 15% better than others. More notably, either F1 or accuracy, the score of our method increases smoothly and almost monotonously when the size of labeled data increases, while other methods shake obviously. It denotes that our model is much stronger and more effective no matter how many data are labeled.

Multiple Relationship Types. We now evaluate the performance of multiple relationship types. Table 3 shows the result of our experiment on both company affiliation and domain affiliation in Weibo data, where our model(S) denotes using just single relationship type, Our model(M) denotes using multiple relationship types, the size of labeled data is set 10%.

As we can see, our method is much better than other methods, although in domain affiliation, libSVM has 1% better than our model in terms of accuracy

Table 3. performance of node classification with multiple relationship types on two tasks in WEIBO data

Task	method	Accuracy	Precision	Recall	F1-score
company affiliation	wvRN	0.278	0.908	0.270	0.416
	CDRN	0.211	**0.931**	0.185	0.308
	libSVM	0.557	0.560	0.593	0.576
	Our model(S)	0.623	0.590	0.842	0.695
	Our model(M)	**0.624**	0.586	**0.887**	**0.705**
domain affiliation	wvRN	0.189	0.776	0.150	0.251
	CDRN	0.468	0.600	0.590	0.593
	libSVM	**0.618**	**0.815**	0.283	0.420
	Our model(S)	0.590	0.547	**0.951**	0.694
	Our model(M)	0.604	0.556	0.947	**0.701**

score, the F1-score of our model is much higher than others (at least 10%). Considering the multiple relation types, the accuracy scores are improved by 0.1% and 1.4% in two tasks while the F1-scores are improved by 1% and 0.7%. It suggests that multiple edge features do add value to our classification model.

5 Conclusion and Future Work

In this paper, we study the problem of node classification in social network, which is an interesting but challenging research domain. We use a factor graph model with semi-supervised learning to infer the category of unlabeled data. In our model, each node in social network is modeled as variable node and various relationships are modeled as factor nodes. In this way, this model can take the advantages of both node features and graph information. Experiments on the different data sets validate the effectiveness of the model we use. It outperforms both model of pure node features (libSVM) and model of pure relationships (wvRN, CDRN).

Node classification in social network is a potential research direction in social network analysis. As future work, since many networks have multiple categories, extending our method to multi-class classification will be quite useful. Considering it is intractable to simply modify the objective function, we should use some indirect approach such as one-against-one to solve the multi-class classification, for example, if there are k categories, we need $k(k+1)/2$ classifiers and vote for each unlabeled node. In addition, the online social network becomes larger and larger, it is interesting to study some fast but effective method for huge networks.

Acknowledgment. The work was Supported by the Upgrading Plan Project of Shenzhen Key Laboratory (No. CXB201005250038A) and The Shenzhen Key Technology R&D Program (No. ZD201010210080A, No.JCYJ20120619153002947). In addition, we thank the anonymous reviewers for their very valuable comments and suggestions.

References

1. Asur, S., Parthasarathy, S.: A viewpoint-based approach for interaction graph analysis. In: Elder IV, J.F., Fogelman-Soulié, F., Flach, P.A., Zaki, M.J. (eds.) KDD, pp. 79–88. ACM (2009)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
3. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks 30(1-7), 107–117 (1998)
4. Desrosiers, C., Karypis, G.: Within-Network Classification Using Local Structure Similarity. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009, Part I. LNCS (LNAI), vol. 5781, pp. 260–275. Springer, Heidelberg (2009)

5. Ebel, H., Mielsch, L.-I., Bornholdt, S.: Scale-free topology of e-mail networks (2002)
6. Gao, B., Liu, T.-Y., Wei, W., Wang, T., Li, H.: Semi-supervised ranking on very large graphs with rich metadata. In: KDD, pp. 96–104 (2011)
7. Heatherly, R., Kantarcioglu, M., Thuraisingham, B.M.: Social network classification incorporating link type values. In: ISI, pp. 19–24. IEEE (2009)
8. Henzinger, M.R., Chang, B.-W., Milch, B., Brin, S.: Query-free news search. In: WWW, pp. 1–10 (2003)
9. Hofmann, T.: Probabilistic latent semantic analysis. In: UAI, pp. 289–296 (1999)
10. Ji, M., Han, J., Danilevsky, M.: Ranking-based classification of heterogeneous information networks. In: KDD, pp. 1298–1306 (2011)
11. Macskassy, S.A., Provost, F.J.: A simple relational classifier. In: Proc. of the 2nd Workshop on Multi-Relational Data Mining, pp. 64–76 (2003)
12. Macskassy, S.A., Provost, F.J.: Classification in networked data: A toolkit and a univariate case study. Journal of Machine Learning Research 8, 935–983 (2007)
13. Murphy, K.P., Weiss, Y., Jordan, M.I.: Loopy belief propagation for approximate inference: An empirical study. In: UAI, pp. 467–475 (1999)
14. Nadakuditi, R.R., Newman, M.E.J.: Graph spectra and the detectability of community structure in networks. CoRR, abs/1205.1813 (2012)
15. Pennacchiotti, M., Popescu, A.-M.: A machine learning approach to twitter user classification. In: Adamic, L.A., Baeza-Yates, R.A., Counts, S. (eds.) ICWSM. The AAAI Press (2011)
16. Spall, J.C.: Estimation via markov chain monte carlo. IEEE Control Systems Magazine 23, 34–45 (2003)
17. Tang, W., Zhuang, H., Tang, J.: Learning to infer social ties in large networks. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011, Part III. LNCS (LNAI), vol. 6913, pp. 381–397. Springer, Heidelberg (2011)
18. Yang, Z., Tang, J., Li, J., Yang, W.: Social community analysis via a factor graph model. IEEE Intelligent Systems 26(3), 58–65 (2011)
19. Zaïane, O.R., Chen, J., Goebel, R.: Mining research communities in bibliographical data. In: Zhang, H., Spiliopoulou, M., Mobasher, B., Giles, C.L., McCallum, A., Nasraoui, O., Srivastava, J., Yen, J. (eds.) WebKDD/SNA-KDD 2007. LNCS, vol. 5439, pp. 59–76. Springer, Heidelberg (2009)