Improving Within-Network Classification with Local Attributes

Sofus A. Macskassy

Fetch Technologies, 2041 Rosecrans Ave, El Segundo, CA 90245
sofmac@fetch.com

Abstract

This paper is about using multiple types of information for classification of networked data in the transductive setting: given a network with some nodes labeled, predict the labels of the remaining nodes. One method recently developed for doing such inference is a guilt-by-association model. This method has been independently developed in two different settings. One setting assumes that the networked data has explicit links such as hyperlinks between web-pages or citations between research papers. The second setting assumes a corpus of non-relational data and creates links based on similarity measures between the instances. Both use only the known labels in the network to predict the remaining labels but use very different information sources. The thesis of of this paper is that if we were to combine the two types of links, the resulting network would carry more information than either type of link by itself. This thesis is tested on six benchmark data sets where we show that this is indeed correct. We further do a sensitivity study on how many links should be created, showing that the combined network gets most of its immediate gain using only a few extra links.

1 Motivation

Recent years have seen a lot of attention on classification with networked data in various domains and settings (e.g., [4, 2, 14, 24]. Networked data is data, generally of the same type such as web-pages or text documents, that are connected via various explicit relations such as one paper citing another, hyperlinks between web-pages, or people calling each other. This paper concerns itself mainly with the problem of *within*-network classification: given a partially labeled network (some nodes have been labeled), label the rest of the nodes in the network.

There have been two separate thrusts of work in this area; one assumes that the data is already in the form of a network such as a web-site, a citation graph, or a calling graph (e.g., [4, 13, 14]. The second area of work has not been cast as a network learning problem, but rather in the area of semi-supervised learning in a transductive setting [1, 10, 25, 2, 24]. These works assume that you are given a corpus of instances and need to first create the links (e.g., given a set of text documents, link them by some similarity score), and then apply classification in this networked data. Interestingly, the same algorithms have been independently developed and used in relational learning [13, 14] as well as in semi-supervised learning [25, 24].

These two existing approaches both ignore information that is readily available. The work on within-network learning has ignored local attributes all together and focused on the univariate case where only the labels are used [14]. Contrast this with the work in the semi-supervised work, where they have no relations and build links using only local attributes (e.g., [25, 24]).

The main thesis of this paper is that augmenting an existing network with links mined from the local attributes ought to increase the information in the network and hence improve the performance of the network classifier. We will show that this is indeed the case on six benchmark data sets, where we augment an existing network by adding K edges from each entity to the K most similar entities to it in the network. We further show that the augmented network is not very sensitivite to K beyond $K = 5$.

We next describe related work, followed by a description of the network classifier and how we augment the network. We then describe our case study in which we test our main thesis, and conclude with a discussion of the results.

2 Related Work

The focus of this paper is on within-network learning, an area that has not yet seen much attention in the relational learning community, with a few exceptions (e.g., [22, 13]. One important aspect of networked data is that it allows *collective inference*, meaning that various interrelated values can be inferred simultaneously. Within-network inference complicates such procedures by pinning certain values, but also offers opportunities such as the application of network-flow algorithms to inference as we describe below. More generally, network data allow the use of the features of a node's neighbors, although that must be done with care to avoid greatly increasing estimation variance and thereby error [9].

Macskassy and Provost [13] investigated a simple univariate classifier, the weighted-vote relational neighbor (wvRN). They instantiated node priors simply by the marginal class frequency in the training data. The wvRN classifier performs relational classification via a weighted average of the estimated class membership scores ("probabilities") of the node's neighbors. Collective inference is performed via a relaxation labeling method similar to that used by Chakrabarti et al. [3]. We use this classifier in our case study.

Relational Bayesian Networks (RBNs, a.k.a. Probabilistic Relational Models [11, 7, 22] were applied in a within-network classification by Taskar et al. [22] to various domains, including a data set of published manuscripts linked by authors and citations. Loopy belief propagation [19] was used to perform the collective inferencing. The study showed that the PRM performed better than a non-relational naive Bayes classifier and that using both author and citation information in conjunction with the text of the paper worked better than using only author or citation information in conjunction with the text.

Recent work outside the area of relational or network learning is directly relevant to within-network classification [1, 10, 25, 24]. These techniques are designed to address semi-supervised learning in a transductive setting [23], but their methods have direct application to certain instances of univariate network classification. Specifically, they consider data sets where labels are given for a subset of cases, and classifications are desired for a subset of the rest. They connect the data into a weighted network, by adding edges (in various ways) based on similarity between cases. We draw upon the work of Zhu et al. [25] and Wang and Zhang [24] below.

3 Classification in networked data

We use an existing and proven method for performing classification of networked data: the weighted-vote relational neighbor (wvRN) [13][1] paired with relaxation labeling (RL) [20, 8] for collective inference. It has been shown that this method is a very strong classifier in networked data [14] (also see [25] and [24]). Using wvRN with an iterative label propagation such as relaxation labeling has been shown to perform better than other collective or exact inference methods [14, 24].

3.1 The weighted-vote Relational Classifier (wvRN)

The wvRN classifier estimates class-membership probabilities based on two assumptions: (1) that the label of a node depends only on its immediate neighbors, and (2) the entities in the graph exhibit homophily—i.e., linked entities have a propensity to belong to the same class (cf. [17]). This homophily-based model is motivated by observations and theories of social networks [17], where homophily is ubiquitous.

Definition. Given $v_i \in V^U$, wvRN estimates $P(x_i|\mathcal{N}_i)$ as the (weighted) mean of the class-membership probabilities of the entities in $\mathcal{N}_i$:

$$P(x_i = X|\mathcal{N}_i) = \frac{1}{Z} \sum_{v_j \in \mathcal{N}_i} w_{i,j} \cdot P(x_j = X|\mathcal{N}_j), \tag{1}$$

where Z is the usual normalizer. As this is a recursive definition (for undirected graphs, $v_j \in \mathcal{N}_i \Leftrightarrow v_i \in \mathcal{N}_j$) the classifier uses the "current" estimate for $P(x_j = X|\mathcal{N}_j)$.

3.2 Relaxation Labeling (RL)

We use relaxation labeling (RL) as described in Macskassy and Provost [14]. Rather than treat G as being in a specific labeling "state" at every point (e.g., as a Gibbs sampler does), relaxation labeling retains the uncertainty, keeping track of the current probability estimations for $\mathbf{x}^U$. The relational model must be able to use these estimations. Further, rather than estimating one node at a time and updating the graph right away, relaxation labeling "freezes" the current estimations so that at step $t+1$, all vertices will be updated based on the estimations from step t. However, doing this often leads to oscillation between states. We therefore use a simulated annealing approach—on each subsequent iteration giving more weight to a node's own current estimate and less to the influence of its neighbors.

More formally, the relaxation labeling inference is defined as:

$$\mathbf{c}_i^{(t+1)} = \beta^{(t+1)} \cdot \text{wvRN}(\mathbf{C}^{(t)}) + (1 - \beta^{(t+1)}) \cdot \mathbf{c}_i^{(t)}, \tag{2}$$

where $\mathbf{c}_i^{(t)}$ is a vector of probabilities (probability distribution) which represents an estimate of $P(x_i|N_i)$ at time step t and wvRN($\mathbf{C}^{(t)}$) denotes applying wvRN using all the estimates from time step t. We define the simulated annealing constants as:

$$\begin{aligned} \beta^0 &= k \\ \beta^{(t+1)} &= \beta^{(t)} \cdot \alpha, \end{aligned} \tag{3}$$

[1] Previously called the probabilistic Relational Neighbor classifier (pRN).

where k is a constant between 0 and 1, which for the case study we set to 1.0, and α is a decay constant, which we set to 0.99. These values were set based on Macskassy and Provost [14].

4 Using local attributes with networked data

In contrast to complex relational learners, wvRN as well as other graph-based methods are *univariate* in that they only consider class label. For classifiers such as these, local attributes are left unused. This is unfortunate as one would think that there is considerable information in the local attributes that should be usable. Macskassy and Provost [15] tried with limited success to make use of local attributes during classification either through setting priors on nodes or using a meta-classifier which combined wvRN with other local-only or relational classifier.

In this paper we take a different approach and borrow techniques from the semi-supervised area (e.g., Wang and Zhang [24]), where the corpus itself is non-relational, but links are created based on local attributes. The classifiers are then used in a within-network setting just as wvRN has been used with the networked data. Wang and Zhang [24] create their edges by calculating similarity scores between instances and using the top-K of such links such that an instance will create K links to the K instances that are most similar to it, using the local attributes.

The idea in this paper is that if we were to augment an existing network with the links created from local attributes, and then applying wvRN-RL, then we will in effect be using both relational as well as local attribute information to predict labels of nodes in the network. The key questions are how we compute similarity scores and what K to use. We answer the former presently and will return to the latter in Section 5.

The data that we consider in this paper is textual in nature (in addition to having explicit links), and we therefore adopt a standard tfidf information retrieval scheme [21]: for each word in the corpus, calculate the tfidf score for that word for a given document, thereby creating a vector of scores for each document. The tfidf score is short for term frequency (tf) inverse document frequency (idf), where tf$(w) = \log(1 + \mathrm{w}_{\mathrm{doc}})$ and idf$(w) = \log(N/N_w)$, where w_{doc} is the number of times word w appears in a given document, N is the size of the corpus and N_w is the number of documents that word w appears in. The similarity of, and the weight of the edge between, two documents is then defined as the cosine of their respective tfidf vectors.

5 Study

The thesis of this paper is that augmenting existing networked data with text-mined links will increase the performance of network classification methods. This case study will empirically test this thesis.

5.1 Data

We use of 6 benchmark data sets from three domains that have been the subject of prior study in machine learning. As this study focuses on combining text-mined links and networked data, instances for which we have no text were removed. Therefore, the statistics we present may differ from those reported previously.

Category	Size
Case Based	432
Genetic Algorithms	512
Neural Networks	1152
Probabilistic Methods	559
Reinforcement Learning	315
Rule Learning	242
Theory	458
Total	3670
Base accuracy	32.39%

Table 1. Class distribution for the CoRA data set.

	Number of web-pages			
Category	**cornell**	**texas**	**washington**	**wisconsin**
course	44	38	77	85
faculty	34	46	31	42
project				3
staff	21	3	10	12
student	128	148	126	156
Total	227	235	244	298
Base accuracy	56.4%	63.0%	51.6%	52.3%

Table 2. Class distribution for the WebKB data set using six-class labels.

CoRA The CoRA data set [16] comprises computer science research papers. It includes the full citation graph as well as labels for the topic of each paper (and potentially sub- and sub-sub-topics).[2] Following a prior study [22], we focused on papers within the machine learning topic with the classification task of predicting a paper's sub-topic (of which there are seven). The class distribution of the data set is shown in Table 1.

Papers can be linked in one of two ways: they share a common author, or one cites the other. Following prior work [12], we link two papers if one cites the other. This number ordinarily would only be zero or one unless the two papers cite each other.

For the text-mined links, we used the abstracts of the papers (we did not have access to the full text of the articles).

WebKB The second domain we draw from is based on the WebKB Project [5].[3] It consists of sets of web pages from four computer science departments, with each page manually labeled into 7 categories: course, department, faculty, project, staff, student or other. As with other work [18, 12], we ignore pages in the "other" category except as described below.

From the WebKB data we produce four networked data sets, one for each of the four universities. Although the data contains six classes, none of them had any text for their department web-pages and only one (Wisconsin) had pages with text from their project pages. In effect, this turned into a four-class problem.

Following prior work on web-page classification, we link two pages by co-citations (if x links to z and y links to z, then x and y are co-citing z) $[3, 12]$. To weight the

[2] These labels were assigned by a naive Bayes classifier [16].

[3] We use the WebKB-ILP-98 data.

Sector	**Number of companies**
Basic Materials	37
Capital Goods	52
Conglomerates	8
Consumer Cyclical	59
Consumer NonCyclical	34
Energy	56
Financial	135
Healthcare	168
Services	275
Technology	402
Transportation	22
Utilities	26
Total	1274
Base accuracy	31.55%

Table 3. Class distribution for the industry data set.

link between x and y, we sum the number of hyperlinks from x to z and separately the number from y to z, and multiply these two quantities. For example, if student x has 2 edges to a group page, and a fellow student y has 3 edges to the same group page, then the weight along that path between those 2 students would be 6. This weight represents the number of possible co-citation paths between the pages. Co-citation relations are not uniquely useful to domains involving documents; for example, for phone-fraud detection bandits often call the same numbers as previously identified bandits (cf. [6]). We chose co-citations for this case study based on the prior observation that a student is more likely to have a hyperlink to her advisor or a group/project page rather than to one of her peers [5]. See Macskassy and Provost [14] for a discussion on edge-selection.

To produce the final data sets, we removed pages in the “other” category from the classification task, although they were used as “background” knowledge—allowing 2 pages to be linked by a path through an “other” page. The composition of the data sets is shown in Table 2.

To create the text-based links, we enhanced the words by tagging special words that appeared in the title, headers or anchors, as this information ought to be useful for deciding similarity.

Industry Classification The final domain we draw from involves classifying companies by industry sector. The Industry Classification data set is based on $38,127$ PR Newswire press releases gathered from April 1, 2003 through September 30, 2003. Each story was tagged with the companies that were mentioned in that story. The data set was then split into two sets: 2809 stories that mentioned more than one company and $35,318$ stories that mentioned only one company.

The former set of 2809 stories was used to create a network of companies where an edge is placed between two companies if they appeared together in the same press release. The weight of an edge is the number of such cooccurrences found in the complete corpus. The resulting network comprises 1274 companies that cooccurred with at least one other company. The latter set of $35,318$ stories was used to create text-mined links. To classify a company, we used Yahoo!’s 12 industry sectors. Table 3 shows the details of the class memberships.

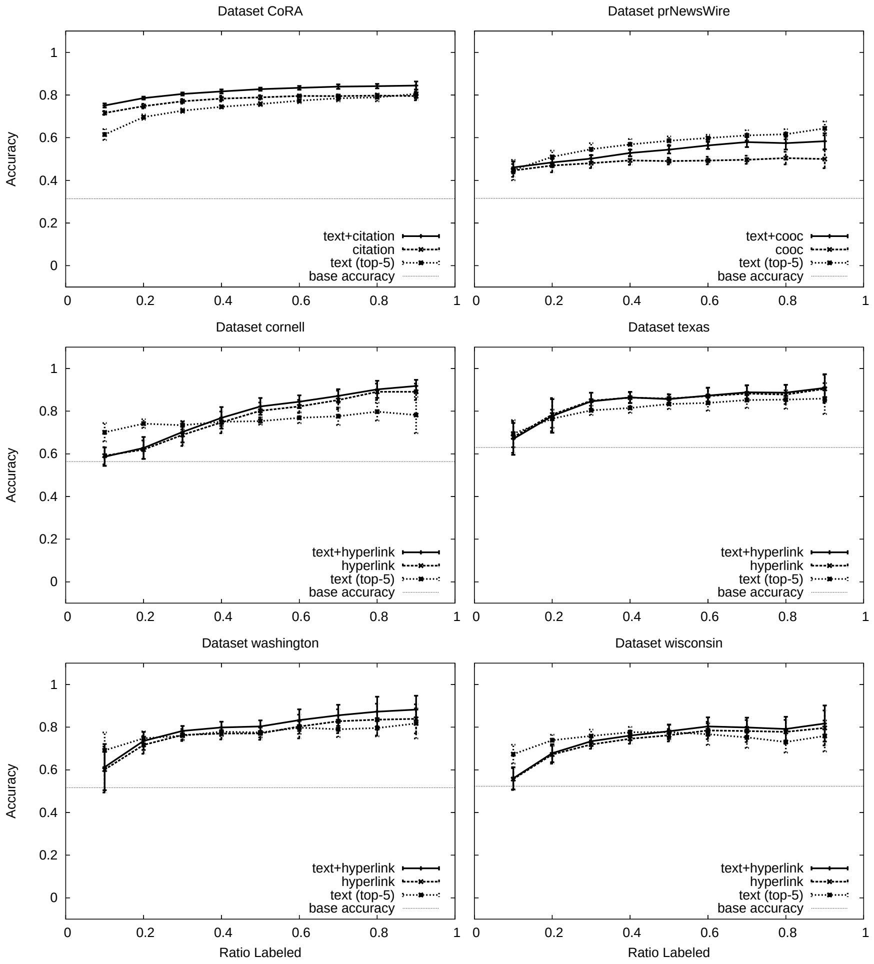

Fig. 1. Comparison of using network-links only versus text-mined links only versus a combination of the two.

5.2 Experimental Methodology

We have for each data set both text-mined links and explicit links between nodes in the network. The question we seek to answer is whether augmenting the explicit links with the text-mined links improve performance. We here use accuracy as the measure of performance, where accuracy is averaged over 10 runs.

We first verify that this is true by comparing the performance of wvRN-RL using only text-mined links versus using only explicit links versus using both types of links together. For the mined links, we set K to 5. In within-network classification, part of the network is initially labeled. We test sensitivity to the type of link by varying from 10% to 90% the amount of initially labeled examples in the network.

Second, we test the sensitivity of K for both the combined network as well as for the text-only network to see how performance changes as K is increased.

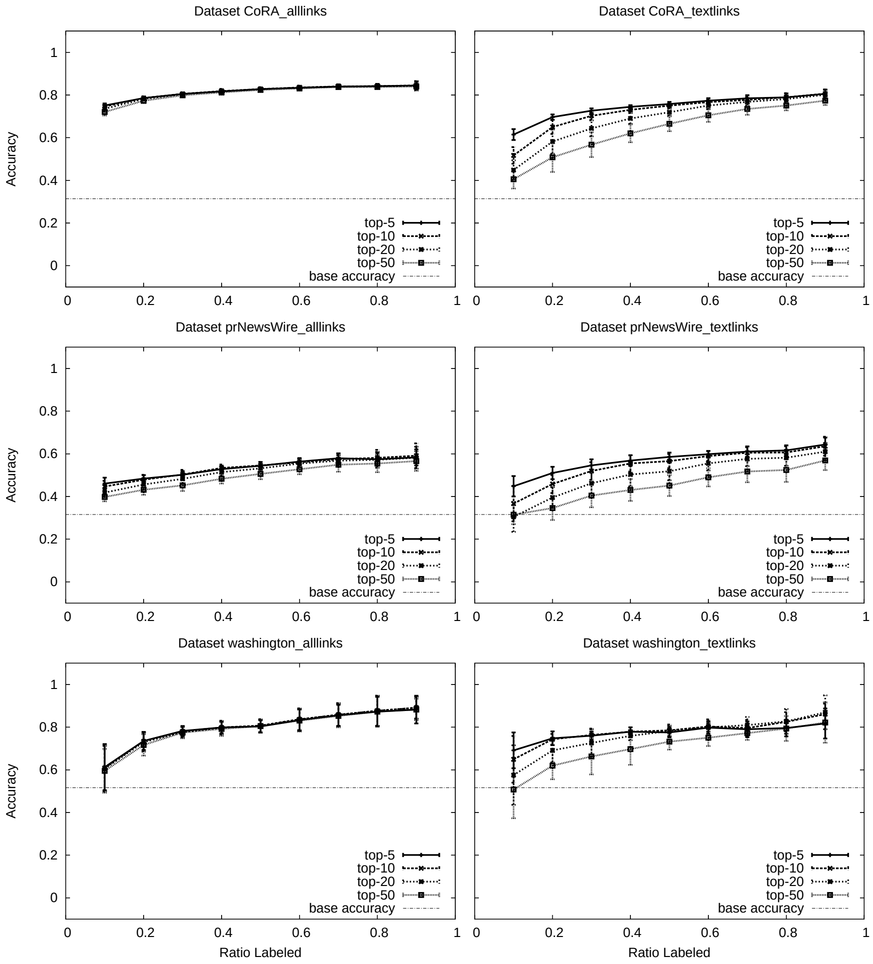

Fig. 2. Testing for sensitivity of K in using the top-K text-mined links. The figure shows a representative selection of the six data sets.

5.3 Results

We first check whether augmenting the network with text-mined links improves performance of wvRN-RL. Figure 1 shows the performance across the six data sets. The results are interesting in three ways: first, we see that augmenting the existing network with text-mined links improved performance on five out of the six data sets. Second, we see that when only a small amount of the network is initially labeled, then using text-mined links alone performed better on three out of the six data sets. Thirdly we see that on five out of the six data sets, when more than half the network is labeled, then the augmented network performed the best and using the text-mined links performed the worst. A paired t-test verifies that these findings are statistically significant at $p < 0.01$. These results clearly show that the thesis of augmenting the network does indeed improve performance and that it never decreases performance. Note that the one case where the augmented network was not as good as the text-only network was the industry sector case where the links were created by implied connections in the text rather than explicitly know links. Therefore, the co-occurrence network may not have been as informative as the one that mined the text.

Second, we investigate the sensitivity to K. Figure 2 shows results on three representative data sets of varying K from 5 to 50 both on the augmented network (on the left) as well as when using only the text-mined links (on the right). Two quite interesting patterns emerge: using $K = 5$ generally performs the best, and using the text-mined links only is more sensitive to K than the augmented network. This argues that when using text-mined links alone, K should be carefully chosen, but in the augmented network this is less critical. Again, the industry classification network was the one exception for the augmented network, perhaps due to the reasons outlined before.

6 Discussion and Limitations

The thesis of this paper was that augmenting networked data with links mined from the local attributes would increase the amount of the information in the network and hence improve the performance of the network classifier.

We described a simple method of adding such links, using the similarity of nodes based on their local attributes as the criterion for adding edges.

We empirically tested our thesis on six data sets, where the local attributes were text, using standard information retrieval measures to calculate similarities between instances. The results clearly show that augmenting the data with these mined links improved performance in the majority of cases and never hurt performance as measured with accuracy.

We further conducted a sensitivity study on how many mined links should be added and found that the augmented network was not very sensitive to this beyond $K = 5$. This, however, may be partly due to how edge-weights were calculated as well as the density of the original network. This is an open question that needs more attention.

This work has shown that augmenting a network can indeed improve performance. This opens the door for many interesting research questions such as what kinds of links we should use to augment the network with. We here proposed a very simple scheme, but it stands to reason that mining for other types of links may very well improve performance even more. This is an edge-selection problem, which is analogous to the feature-selection problem in standard machine learning. A second issue is how the network should be augmented. As mentioned above, there are two obvious issues: (1) The weight of the edges: if the edges in the original network have very large weights, then the weight of the augmented links may need to be scaled up accordingly; and (2) the number of edges to add. If the graph is very sparse, then adding too many new edges may unduly favor the mined links over the existing links. The opposite case is also true, if you have a dense graph and only add a few links.

References

1. Blum, A., Chawla, S.: Learning from Labeled and Unlabeled Data using Graph Mincuts. In: Proceedings of the International Conference on Machine Learning (ICML). (2001) 19–26
2. Blum, A., Lafferty, J., Reddy, R., Rwebangira, M.R.: Semi-supervised learning using randomized mincuts. In: Proceedings of the 21st International Conference on Machine Learning (ICML). (2004)

3. Chakrabarti, S., Dom, B., Indyk, P.: Enhanced Hypertext Categorization Using Hyperlinks. In: ACM SIGMOD International Conference on Management of Data. (1998)
4. Cortes, C., Pregibon, D., Volinsky, C.T.: Communities of Interest. In: Proceedings of Intelligent Data Analysis (IDA). (2001)
5. Craven, M., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., Quek, C.Y.: Learning to Extract Symbolic Knowledge from the World Wide Web. In: 15th Conference of the American Association for Artificial Intelligence. (1998)
6. Fawcett, T., Provost, F.: Adaptive fraud detection. Data Mining and Knowledge Discovery **3** (1997) 291–316
7. Friedman, N., Getoor, L., Koller, D., Pfeffer, A.: Learning Probabilistic Relational Models. In: Sixteenth International Joint Conference on Artificial Intelligence (IJCAI). (1999)
8. Hummel, R.A., Zucker, S.W.: On the foundations of relaxation labeling processes. IEEE Transactions on Pattern Analysis and Machine Intelligence **5**(3) (1983) 267–287
9. Jensen, D., Neville, J., Gallagher, B.: Why Collective Inference Improves Relational Classification. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2004)
10. Joachims, T.: Transductive Learning via Spectral Graph Partitioning. In: Proceedings of the International Conference on Machine Learning (ICML). (2003)
11. Koller, D., Pfeffer, A.: Probabilistic Frame-Based Systems. In: AAAI/IAAI. (1998) 580–587
12. Lu, Q., Getoor, L.: Link-Based Classification. In: Proceedings of the 20th International Conference on Machine Learning (ICML). (2003)
13. Macskassy, S.A., Provost, F.: 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)
14. Macskassy, S.A., Provost, F.: Classification in Networked Data: A toolkit and a univariate case study. Technical Report CeDER Working Paper 04-08, Stern School of Business, New York University (2004) Revised June, 2006.
15. Macskassy, S.A., Provost, F.: Suspicion scoring of entities based on guilt-by-association, collective inference, and focused data access. In: Annual Conference of the North American Association for Computational Social and Organizational Science (NAACSOS). (2005)
16. McCallum, A., Nigam, K., Rennie, J., Seymore, K.: Automating the Construction of Internet Portals with Machine Learning. Information Retrieval **3**(2) (2000) 127–163
17. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a Feather: Homophily in Social Networks. Annual Review of Sociology **27** (2001) 415–444
18. Neville, J., Jensen, D., Friedland, L., Hay, M.: Learning Relational Probability Trees. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2003)
19. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann (1988)
20. Rosenfeld, A., Hummel, R., Zucker, S.: Scene labeling by relaxation operations. IEEE Transactions on Systems, Man and Cybernetics **6** (1976) 420–433
21. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc. (1983)
22. Taskar, B., Segal, E., Koller, D.: Probabilistic Classification and Clustering in Relational Data. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI). (2001) 870–878
23. Vapnik, V.N.: Statistical Learning Theory. John Wiley, NY (1998)
24. Wang, F., Zhang, C.: Label propagation through linear neighborhoods. In: Proceedings of the 23rd International Conference on Machine Learning (ICML). (2006) 985–992
25. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In: Proceedings of the 12th International Conference on Machine Learning. (2003)