Semi-Supervised Classification of Network Data Using Very Few Labels

Frank Lin and William W. Cohen

CMU-LTI-09-017

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
www.lti.cs.cmu.edu

© 2009, Frank Lin and William W. Cohen

Semi-Supervised Classification of Network Data Using Very Few Labels

Frank Lin* William W. Cohen*

Abstract

The goal of semi-supervised learning methods is to reduce the amount of labeled training data required by learning from both labeled and unlabeled instances. We make contribution towards this goal along several dimensions.

Macskassy and Provost [13] proposed the weighted-vote relational neighbor classifier (wvRN) as a simple yet solid baseline for semi-supervised learning on network data. It is shown to be essentially the same as the Gaussian-field classifier proposed by Zhu et al. [22] and proves to be very effective on many benchmark network datasets.

We describe another simple and intuitive semi-supervised learning method based on random graph walk that outperforms wvRN by a large margin on several benchmark datasets when very few labels are available.

Secondly, we show that using authoritative instances as training seeds — instances that arguably cost much less to label — dramatically reduces the amount of labeled data required to achieve the same classification accuracy. For some existing state-of-the-art semi-supervised learning methods the labeled data needed is reduced by a factor of 50.

Third, we offer insights as to why learning methods based on random graph walk are able to more fully exploit the unlabeled data than previous methods.

Based on the above observations, we strongly recommend the proposed method as a strong baseline for future research on semi-supervised classification of network data.

1 Introduction

Traditional machine learning or *supervised learning* methods for classification require large amounts of labeled training instances, which are often difficult to obtain. In order to reduce the effort required to label the training data, two solution have been proposed: *semi-supervised learning* methods [21] and *active learning* methods (e.g., [5, 23]). Semi-supervised learning methods learn from both labeled *and* unlabeled instances, reducing the amount of labeled instances needed to achieve the same level of classification accuracy. Active learning methods reduce the number of labels required for learning by intelligently choosing which instances to label first.

The field of semi-supervised learning has been very active in recent years, and many of the learning methods proposed fall under the category of *graph-based semi-supervised learning* [21, 17, 22, 20, 8, 13], which views the instance space as a graph where instances are the nodes and similarities between the instances define weighted edges. This representation is powerful and exciting; almost any dataset can be represented as a graph and many graph algorithms and theories can be applied.

However, as the number of proposed methods increases over the years, many questions remain unanswered. How do these methods relate to each other? Which methods do better, under what condition, and on what type of data? How is one method better than the other and, more importantly, why is it better? What method should be use as a strong baseline when working on a particular type of data?

In this work, we aim to address some of these questions. We are especially interested in reducing labeling cost, so we focus on cases where there are very few labels.

First, we describe a semi-supervised learning method based on random graph walk and relate it to methods that fall under the class of graph walk-based algorithms, such as [20, 6, 8]. The core computation of these methods usually involve finding the dominant eigenvector of some form of affinity matrix or transition matrix of the graph. The proposed method is probably is simplest of them all, yet it is also intuitive and extremely effective and captures the power of these graph walk-based methods.

Second, in the quest for reducing the cost of obtaining instance labels, one issue has not been considered in prior work: that in many practical settings, *some instances are easier to label* than others. For example, in classifying websites, a better-known website is very likely easier for a domain expert to label, since the expert would be more likely to be familiar with it, and since the website would be less likely to be difficult-to-evaluate because of having content that is limited (or simply incoherent). Similarly, in classifying technical papers, the more influential papers should be easier for a subject expert to label.

In selecting seeds (as labeled training data), we evaluate using highly authoritative instances in addition to being arguably easier to label, these authoritative instances are arguably more likely to spread their influence (and their labels) to their neighbors, therefore making them better seeds (labeled instances) in a semi-supervised learning setting.

We test the proposed methods on five network

*School of Computer Science, Carnegie Mellon University

datasets (i.e., data in the form of a graph, where each node is a learning instance) and show the proposed methods outperforms some existing semi-supervised learning methods [22, 13] by a large margin when the number of labeled instances is small. In addition, the classification performance is competitive with some state-of-the-art *fully supervised* methods for learning in graphs [11, 10] — a surprising result given that many fewer labels are used, and the methods we propose (as currently implemented) makes no use of the "content" of the instances, only the graph structure.

Third, perhaps most importantly, based on the experimental results, we offer intuition and explanations as to why random graph walk-based methods outperform some of the existing methods and point out what may hinder these methods from fully exploiting unlabeled data when very few labeled training instances are given.

Lastly, we strongly recommend the proposed graph walk-based method as a baseline for future research in semi-supervised learning. In addition to its high classification accuracy, the method is simple to implement and is based on a family of well-studied algorithms (random graph walks), and it is also highly scalable, requiring time linear in the number of edges of the graph.

2 The MultiRankWalk Algorithm

The proposed semi-supervised learning method is based on random graph walk. Its basic component is similar to PageRank [15], personalized PageRank [7], and random walk with restart (RWR) [18]. In general, given a graph $G = (V, E)$, random walk algorithms such as the three mentioned above return as output a ranking vector $\mathbf{r}$ satisfying the following equation:

$$\mathbf{r} = (1 - d)\mathbf{u} + dW\mathbf{r} \tag{2.1}$$

where W is the weighted transition matrix of graph G where transition from i to j is given by $W_{ij} = 1/degree(i)$. $\mathbf{u}$ is a normalized teleportation vector where $|\mathbf{u}| = |V|$ and $||\mathbf{u}||_1 = 1$. d is a constant damping factor. The ranking vector $\mathbf{r}$ can be solved for by finding the dominant eigenvector of $(1 - d)(I - dW)^{-1}\mathbf{u}$ or iteratively substituting $\mathbf{r}^t$ with $\mathbf{r}^{t-1}$ until $\mathbf{r}^t$ converges. Equation 2.1 can be interpreted as the probability of a random walk on G arriving at node i, with teleportation probability $(1 - d)$ at every step to a node with distribution $\mathbf{u}$. For later use we will define the ranking vector $\mathbf{r}$ as a function of G, $\mathbf{u}$, and d: $\mathbf{r} = RandomWalk(G, \mathbf{u}, d)$.

The difference between some of the different random walk algorithms lies in the use and interpretation of $\mathbf{u}$. In PageRank [15], where G is a network of hyperlinked web pages, $\mathbf{u}$ is simply a uniform vector; with probability $1 - d$ a web surfer gets tired of following the links he sees and jump to a random page. In personalized PageRank [7], each web surfer, instead of jumping to a random page, jumps to a page according to his or her unique preference, the preference encoded as a normalized distribution $\mathbf{u}$. In random walk with restart, $\mathbf{u}$ is an all-zero vector except for $\mathbf{u}_i = 1$ where i is the *starting node*; at every time step the random walker follows an edge with probability d or jumps back to i (*restarts*) with probability $1 - d$.

In our proposed method, the graph G describes data in a classification learning framework: the nodes are instances and edges represent similarity or relations between the instances. Labeled training instances of each class is described by a vector $\mathbf{u}$, the *seed vector*, where each non-zero element corresponds to a labeled training node. The random walk describes classification as a process of finding similar instances based on citation or recommendation of the current instance. For each class c, at every time step the process may follow a recommendation with probability d or it may decide to start the process again at an instance labeled c with probability $1 - d$. The process is repeated for every class and the class of an unlabeled instance is decided by which class c's process visited the instance most often. The learning algorithm is formally described in Figure 1.

Given: A graph $G = (V, E)$, corresponding to nodes in G are instances X, composed of unlabeled instances X^U and labeled instances X^L with corresponding labels Y^L, and a damping factor d.

Returns: Labels Y^U for unlabeled nodes X^U.

For each class c

1. Set $\mathbf{u}_i \leftarrow 1, \forall Y_i^L = c$
2. Normalize $\mathbf{u}$ such that $||\mathbf{u}||_1 = 1$
3. Set $R_c \leftarrow RandomWalk(G, \mathbf{u}, d)$

For each instance i

- Set $X_i^U \leftarrow argmax_c(R_{ci})$

Figure 1: The MultiRankWalk algorithm.

We will refer to this classification algorithm as **MultiRankWalk**, because it computes **Multi**ple **Rank**ings using random **Walk**s. Although we developed it independently, this is similar to previously described meth-

ods [20, 6]. The additional contributions of this paper are 1) more datasets, many of them used evaluating other semi-supervised methods, are used; 2) comparison is made to other baselines, including Gaussian-fields classifier, stacked learning, and spectral clustering methods; 3) seed selection in a semi-supervised setting; 4) focus on small number of seeds; and 5) the analysis of its relation to and difference from wvRN and Gaussian-fields classifier.

3 Related Work

3.1 Graph Walk-based SSL Methods The idea of using random walks to propagate labels from labeled nodes to unlabeled nodes in graph is not new; for example, the local and global consistency method [20] have at its core iterating the equation $F(t+1) = \alpha SF(t) + (1-\alpha)Y$ until convergence for each class, where F, S, α, and Y are analogous to r, W, d, and u in Equation 2.1; and in [6] the resulting rank vectors are used as features in a SVM classifier. Perhaps less obviously, the conditional likelihood component in [8] can also be seen as random graph walks, with the difference that the walks do not restart (no damping factor) and a heuristic stopping criterion is used instead of convergence.

As with many other semi-supervised learning methods, in using graph walk-based methods we make the assumption that the instance graph is *homophilous* — i.e., that instances belonging to the same class tend to link to each other or have higher edge weight between them. A homophilous instance graph can be constructed using similarity functions on instance features, but it is also found in many naturally occurring networks—including networks of websites, blogs, and paper citations [2] — and often arises when a single network is jointly constructed by several communities.

With graph-based approaches comes the question of how the graph is constructed. When instances are not explicitly linked to each other, usually a similarity function is applied to local features of each pair of instances to derive weighted edges between them [22]. When instances are explicitly linked to each other, such as a network of websites connected by hyperlinks, the edges simply correspond to the binary presence of a link (or are weighted by the number of links between two instances). In many datasets, hybrid approaches are used when both local features and explicit links are available [12].

The advantage of feature-derived edges is that they can be potentially used on almost any data without explicit links. However, the algorithm could be sensitive to the similarity function and the similarity function may require re-engineering when the same algorithm is applied to a different dataset. In addition, since the similarity function is applied to all pairs of instances, the graph might be very dense, incurring a heavy computation cost. Having explicit, naturally occurring links means a lower computation cost and there is no need to engineer similarity functions; however, not all data comes with explicit links.

In this work we will focus on network datasets with explicit links as edges.

3.2 Other Related Work Active learning methods [5, 19, 23] aim also to reduce the number of labels required by computing which instance, when the label is known, will best help them in classifying the rest of the data. Active learning is usually done in an interactive setting, where the algorithm selects an instance, and then user labels the instance. This process is repeated, with the classification accuracy going up at each iteration (hopefully more than if the selected instances were chosen at random). There is an important difference between active learning methods and our proposed method of selecting seeds: with our proposed method the calculation is done once at the beginning instead of in every round of learning.

Classification of data in graphs, called *collective classification* or *relational learning* (e.g. web page classification [4, 3] and scientific paper classification [11, 10]), has also been studied in a more traditional machine learning setting. Although these method do make use of the explicit links in the data, they are supervised learning methods and still require a large amount of labeled data. In this paper we will compare our semi-supervised method against of these supervised methods.

Clustering, or *unsupervised learning* methods are similar to graph-based semi-supervised learning methods in that they usually rely on similarity functions or the graph structure of the data to cluster instances into k clusters, k being a pre-specified parameter or a learned threshold based on another parameter. A class of these methods, called spectral clustering methods [16, 14], are similar to our approach in that the algorithms have a direct random graph walk interpretation. We will also compare our method with some spectral clustering methods on some two-class datasets.

4 Seed Selection

Semi-supervised learning methods require labeled training instances as*seeds*, and we propose using more authoritative instances. There are two advantages to prefer highly authoritative instances as seeds:

First, popular or authoritative instances are easier to obtain labels for because a) domain experts are

more likely to recognize them and label them without comprehensive assessment of the instances, resulting in less time and human effort spent, and b) popular or authoritative instances are more likely to have already labels available. As an example, websites such as *www.etalkinghead.com* contain blog directories that are organized according to political leaning. Although the directories contain only a small percentage of all the political blogs out there (150 liberal, 148 conservative, and 48 libertarian blogs as of the time of this writing), these are likely highly popular and authoritative blogs and can be used as seed instances.

Second, popular or authoritative instances will likely to have many incoming links (other instances are more likely to link to or cite them) and sometimes outgoing links as well (in the case of blogs, popular blogs are usually well-kept and contain more entries and links). Having many incoming and outgoing links helps to propagate the labels faster and more reliably when using a graph-based semi-supervised learning method such as [13] and the MultiRankWalk proposed in the previous section.

Based on these assumptions, we propose a general seeding method to test our hypothesis: **ranked-at-least-n-per-class**. This method takes as input a ranked list of instances according to a preference measure, the most preferred instance on top. Given a number n, we start at the top and label each instance as a seed instance going down until we have at least n seeds per class labeled as seed instances. This method simulates a domain expert labeling a given list of instances (ordered according to some preference measure) and labeling instances one by one until he or she feels an adequate number of instances have been labeled for each class. In addition, this seeding method makes sure there is at least one instance of every class in the training data while preserving a natural labeling process (as opposed to a class-stratified training data that gives the classifier perfect prior probabilities).

For all experiments in this work we vary n and test these different seed ranking preferences:

Random seeding is a baseline measure that randomly orders the list of instances.

Link Count seeding ranks the instances based on the number of edges connected to it; instances with more connecting edges are preferred.

PageRank seeding ranks the instances based on PageRank [15]; nodes with higher PageRank scores are ranked preferred.

4.1 Datasets To assess the effectiveness of our method, we test it on five different datasets. The first three datasets are from the political blog domain, which we refer to as the UMBCBlog dataset, the AGBlog dataset, and the MSPBlog dataset. The fourth and fifth datasets are from the scientific paper citation domain, which we refer to as the Cora dataset and the CiteSeer dataset. All of these datasets have in common that they contain explicit links between the instances in the form of hyperlinks or citations. In constructing the graph from these datasets, we take the simplest approach possible; in each case the resulting graph contains only undirected, unweighted edges. Some statistics of the five datasets can be find in Table 1.

The **UMBCBlog** dataset is constructed in the same way as Kale et al. did in [9]: first we find a set of overlapping blogs between the ICWSM 2007 BuzzMetrics [1] dataset and Lada Adamic's labeled dataset [2], then we generated a graph using links found in the BuzzMetrics dataset posts, and lastly we take the largest (weakly) connected component of the graph and end up with a dataset of 404 connected blogs that are labeled either *liberal* or *conservative*.

The **AGBlog** dataset is constructed by creating a graph from Lada **A**damic and Natalie **G**lance's political **blog** dataset [2] and taking the largest connected component. This dataset contains 1222 connected blogs. Every blog within this datasets are labeled either *liberal* or *conservative*. Although the UMBCBlog and AGBlog datasets are not entirely unrelated (the blogs from the UMBCBlog dataset are mostly a subset of the blogs from the AGBlog dataset), using the above procedure we effectively create two distinct datasets of different size and link structure. In the UMBCBlog dataset, the links are gathered around May 2006 from the content of the blog posts; in the AGBlog dataset, the links are from two months before the 2004 presidential election and are extracted mostly from the sidebars of the blogs [2]. The links from the UMBCBlog dataset can be considered more *transitory*, pertaining to the blogger's interests at the time of the post, while links from the AGBlog dataset can be considered more *stationary*, indicating the blogger's long-term interests and recommendations. In addition, it should be pointed out that the labeling of the political blog datasets is not 100% accurate as noted in [2].

The **MSPBlog** dataset is provided by the researchers at Microsoft Live Labs and is constructed separately from the above two datasets. From a large collection of automatically crawled news and blog sites, political ones are manually labeled either as *liberal* or *conservative*. Again, the largest connected component from the labeled subgraph is used, resulting in 1031 connected pages. Unlike the previous two datasets, the distribution of classes for MSPBlog dataset is much more skewed. Class label distribution for the three political

blog datasets can be found in Table 2.

The other two datasets are scientific paper citation datasets. The **Cora** dataset contains 2708 papers from 7 categories and the **CiteSeer** dataset contains 3312 papers from 6 categories. The class names and class label distributions for these two datasets can be found in Table 3 and the details of their construction is described in [11]. Since our algorithm takes only the link structure and a small number seed nodes as input, we require the graph to be connected. Again, we extract the largest connected component from these datasets and end up with 2485 papers for the Cora dataset and 2110 papers for the CiteSeer dataset. An edge exist in the graph between node a and node b if paper a cites paper b or vice versa.

	Nodes	Edges	Density
UMBCBlog	404	2725	0.01670
AGBlog	1222	19021	0.01274
MSPBlog	1031	9316	0.00876
Cora	2485	5209	0.00084
CiteSeer	2110	3757	0.00084

Table 1: Some statistics on the four datasets. Density is the ratio of number of edges to the number of nodes squared.

	UMBCBlog	AGBlog	MSPBlog
Liberal	198	586	375
Conservative	206	636	656

Table 2: Class distribution for the political blog datasets.

Cora		CiteSeer	
Neural Networks	726	HCI	304
Case Based	285	IR	532
Reinforcement Learning	214	Agents	463
Probabilistic Methods	379	AI	115
Genetic Algorithms	406	ML	308
Rule Learning	131	DB	388
Theory	344		

Table 3: Class distribution for the citation datasets.

5 Experiments and Results

Description of experiments and discussion of results are divided into three subsections. In the first subsection, we compare our random walk learning method, MultiRankWalk, against Zhu [22] and Macskassy's [13] semi-supervised learning algorithm on five network datasets and the effect of different seed preferences on these learning algorithms. In the second subsection, we compare MRW with graph-based clustering methods and a state-of-the-art collective classification algorithm. In the last subsection, we vary the parameter d of MRW and observe its effect on classification performance.

In all classification performance figures, we vary the number of labeled instances by changing the seeding parameter n mentioned in Section 4. For UMBCBlog, AGBlog, Cora, and CiteSeer, we use $n = 1$, 2, 5, 10, 20, and 40; for MSPBlog, we use an additional $n = 80$. Note that in the figures to follow x-axis refers to the number of labels and not n. In each experiment, all instances not used as labeled training instances is used as test data; so as the amount of training data increases the number of test data decreases. The reported numbers when random seeding is involved are averaged over 20 runs each.

5.1 Comparing Semi-Supervised Algorithms and Seed Preferences The weighted-vote relational neighbor classifier (wvRN) [13] is a simple label propagation algorithm that estimates class-membership probabilities by assuming the existence of homophily. It is one of the best classifiers on many benchmark network datasets and as noted by Macskassey and Provost in [13], Zhu's Gaussian field classifier [22] is "essentially identical" to wvRN except with a principled semantics and exact inference. We compare MultiRankWalk's performance with Zhu and Macskassy's algorithm on the five network datasets using different seed preferences.

Due to space limitations, we show only the macro-averaged F1 score instead both the accuracy and the macro-averaged F1 score. The accuracy (ratio of correctly labeled test instances to the total number of test instances), though not shown, are always higher than the F1 score in the experiments we ran. The macro-averaged F1 score is defined as $\frac{1}{N}\sum_{c=1}^{N}\frac{2P_cR_c}{P_c+R_c}$ where N is the number of classes and P_c is the precision of the classifier for class c and R_c is the recall of the classifier for class c. The macro-averaged F1 score is usually preferred when the class label distribution is unbalanced, which is true of most of the datasets presented here. Figure 2 shows the classification performance on the five datasets with the three seeding algorithms; the rows are the different datasets and the columns are the different seed preferences. RMW is compared against wvRN in each chart, and the algorithm that significantly (with p¡0.001) outperforms the other at a particular amount of labeled data is indicated by a box around the point. Details of the significance tests will be described later.

We make a few observations in this figure. First, MRW is able to achieve high classification accuracy with very few labeled instances. The first point on the charts shows that on UMBCBlog and AGBlog, MRW achieves F1 score of above 0.9 on with just two labeled instances

(training data size is 0.5% and 0.16% of test data size, respectively). On MSPBlog MRW achieves F1 scores close to 0.9 with just three or four labeled instances (0.3% of test data size). On the seven-class Cora dataset MRW achieves scores above 0.6 with about 20 labeled instances and on the six-class CiteSeer dataset MRW achieves 0.5 with about 30 labeled instances (0.8% and 1.4% of test data size, respectively).

Second, on most datasets and seed preferences MRW outperforms wvRN by a large margin when the amount of training data is very small. The only exception to this is the CiteSeer dataset when wvRN is paired with **LinkCount** or **PageRank** seeding — MRW, though still better than wvRN with about 30 or 60 seeds, it is not significantly so, and wvRN significantly outperforms MRW from 100 to 250 seeds. On all datasets MRW and wvRN F1 scores converge when the training data size reaches above 30% of the test data size.

Third, the performance difference between MRW and wvRN is the greatest when seeds are chosen randomly. This suggests that MRW is more robust to varying quality of the labeled data.

Figure 3 and Figure 4 shows the classification performance on the five datasets with the two algorithms, this time the different seed preferences are compared against each other on the charts. If at a particular point, either LinkCount or PageRank seed preference significantly (with pj0.05) outperforms Random seeding, a box is put around the point. Note that points are not aligned exactly due to the **ranked-at-least-n-per-class** seeding described in section 4.

With wvRN, we see that preferring more authoritative seeds dramatically outperforms random seeds, especially when the number of labeled instances is small; on the blog datasets preferring more authoritative seeds reduces the amount of labeled instances required to reach the same level of performance by a factor of 40 to 50! Out of the two authority-based preferences, PageRank seems to be a little better and more consistent in yielding quality seeds, as seen in the first few points of UMBCBlog, AGBlog, and CiteSeer datasets.

With MRW, the difference between random seeds and the authoritative seeds are not as dramatic, one reason being that on the political blog datasets the F1 is already very high with random seeding. However, a significant difference is still observed on AGBlog, MSPBlog, and Cora datasets when the number of labeled instances is very small. When comparing LinkCount and PageRank, again we see PageRank a better and more stable seed preference, and the performance of different seed preferences converge when training data is large enough.

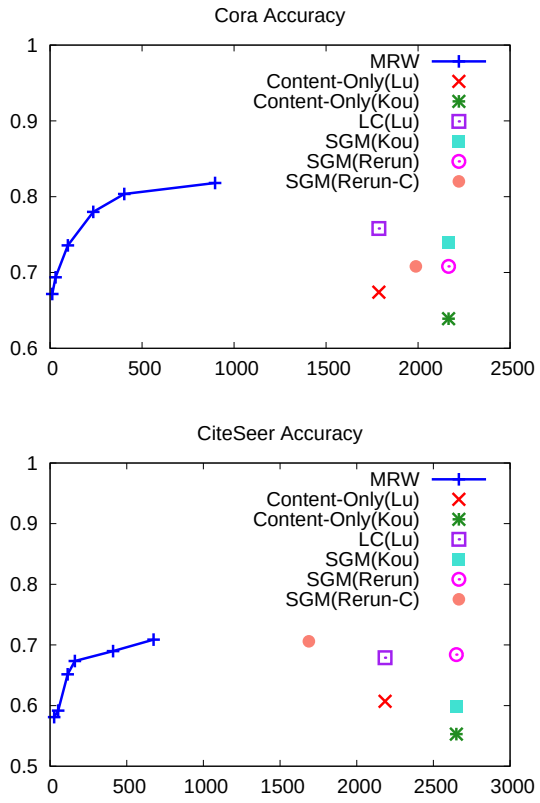
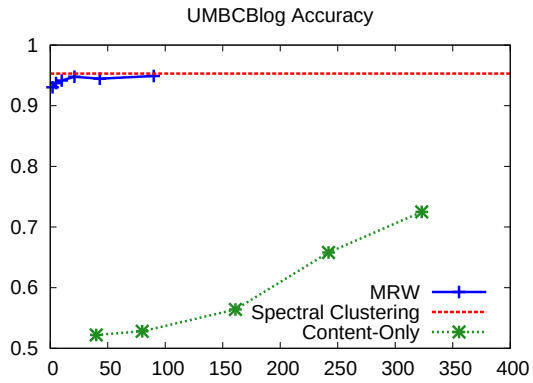


Figure 5: Citation datasets results compared to supervised relational learning methods. The x-axis indicates number of labeled instances and y-axis indicates labeling accuracy.

5.2 Versus Supervised Relational Learning and Spectral Clustering Methods To show how much classification power can be gained from link structure alone, we compare the results of our algorithm against some supervised relational learning methods, shown in Figure 5. The numbers shown in the charts are accuracy scores. The algorithms are labeled on the figures as follows: **MRW** is MultiRankWalk algorithm using PageRank seeds; **Kou** is the best result reported in [10]; **Kou-Rerun** is the result from our re-run of **Kou**; **Kou-Rerun-C** is our re-run of **Kou** using the connected version of the dataset; **Lu** is the best result reported in [11]; **Content-Kou** is the content-only baseline reported in [10]; and **Content-Lu** is the content-only baseline reported in [11]. MRW shows outstanding performance considering the simplicity of the algorithm, the small number of labeled instances required, and using only the link structure. Following [13] we recommend label propagation algorithms such as MRW as a strong baseline for semi-supervised learning or supervised relational learning for network data.

MRW is based on random walks on graphs, and its strong performance on the political blog datasets



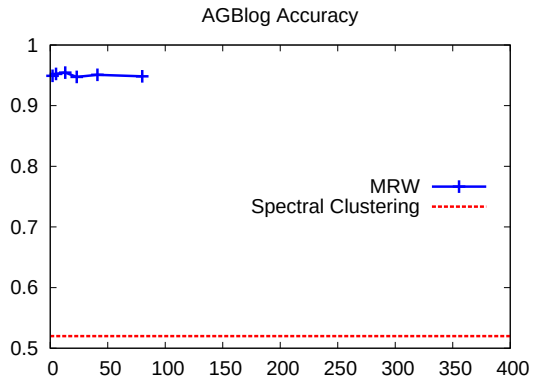
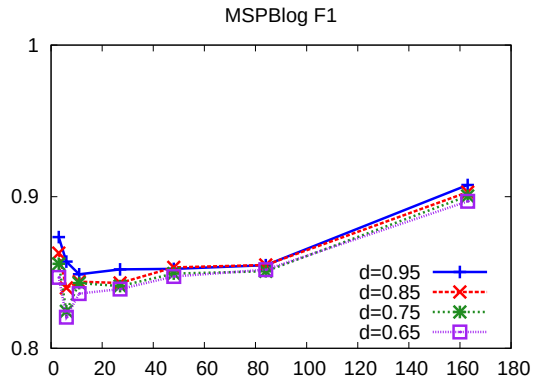


Figure 6: Two political blog datasets results compared to spectral clustering methods. The x-axis indicates number of labeled instances and y-axis indicates labeling accuracy.



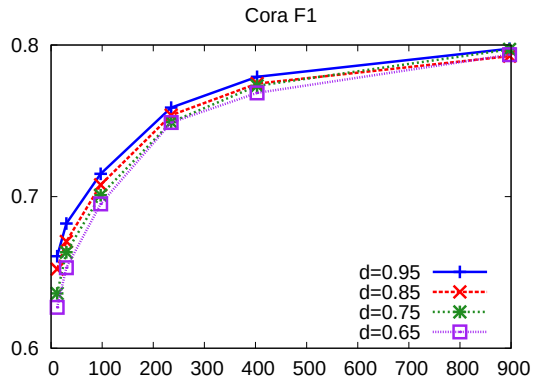


Figure 7: Results on three datasets varying the damping factor. The x-axis indicates number of labeled instances and y-axis indicates labeling macro-averaged F1 score.

may make one wonder if these datasets are also easily divided into two groups by spectral clustering methods — which also have direct random walk interpretations — without using any labeled data. The results of MRW compared against two spectral clustering algorithms are shown in Figure 6. The numbers shown in the charts are accuracy scores. The algorithms are labeled on the figures as follows: **MRW** is MultiRankWalk algorithm using PageRank seed preference; **Spectral Clustering** is both the spectral clustering algorithm proposed by Ng et al. [14] and Normalized Cuts [16] — they have the exact same performance on these two datasets. **Content-Only** is the content-only Naïve Bayes using bag-of-words features, shown here for comparison; the AGBlog dataset does not have one due to its lack of content data. The results show that spectral clustering methods were indeed able to cluster the two classes as well as MRW on UMBCBlog, they failed to do so completely with AGBlog. We did further experiments to investigate why this was the case, but will only include it in future work due to space limitations.

5.3 Damping Factor The effect of the damping factor d on the proposed learning method is shown in Figure 7; the general trend is that a higher damping factor consistently result in slightly better classification performance. This suggests that it is important for the algorithm to propagate the labels further by not "damping" the walk too much, especially when the number of labeled instances is small.

5.4 Significance Tests For comparing significant difference between wvRN and MRW when using CountLink and PageRank seed preferences, a one-tail paired McNemar's test on the classification result of individual instances is used with $p < 0.001$ reported as significant. For comparing significant difference between wvRN and MRW when using Random seed preference, the 20 accuracy scores from the 20 random trials are used in a one-tail Mann-Whitney U test with $p < 0.001$ reported as significant. For comparison between the random seeding and authority-based seed preferences, the classification result of individual instances is used in a one-tail Mann-Whitney U test with $p < 0.05$.

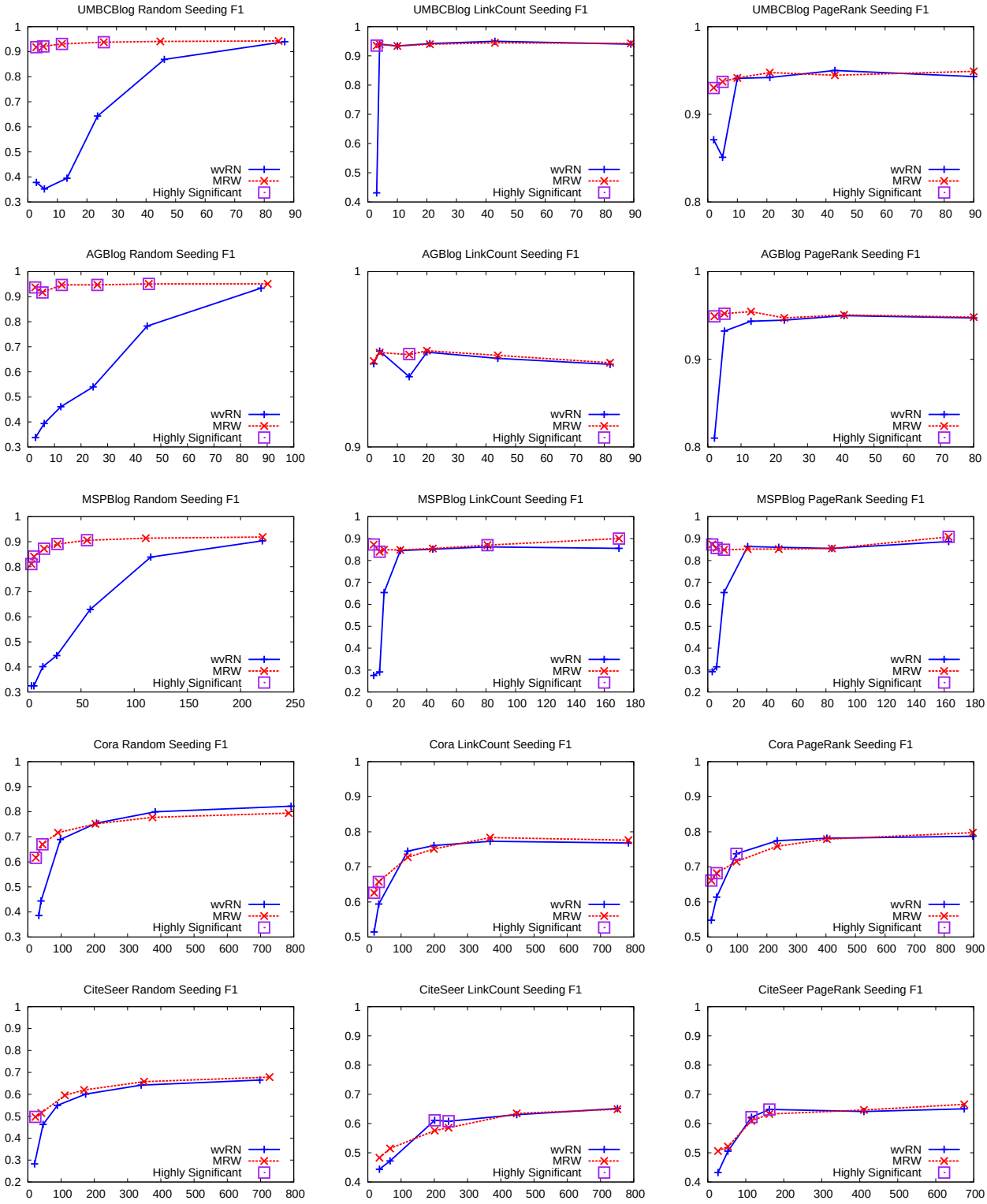


Figure 2: All five datasets results varying the learning algorithm. The x-axis indicates number of labeled instances and y-axis indicates labeling macro-averaged F1 score. Square block around a point indicates statistical significance with $p < 0.001$.

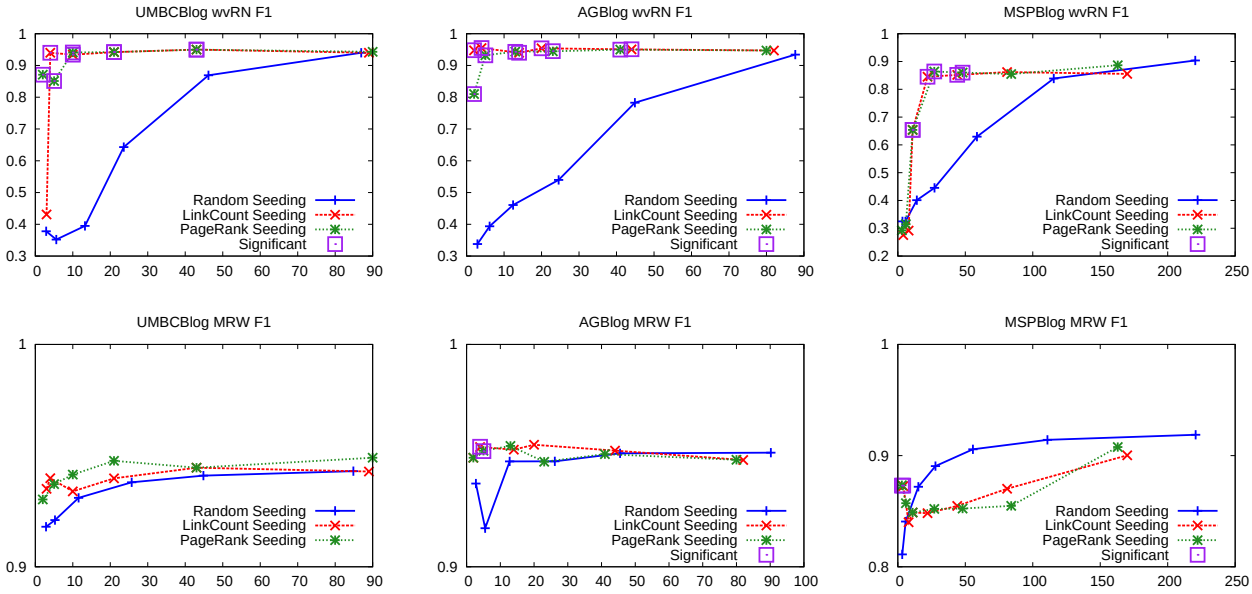


Figure 3: Political datasets results varying the seeding method. The x-axis indicates number of labeled instances and y-axis indicates labeling macro-averaged F1 score. Square block around a point indicates statistical significance with $p < 0.05$.

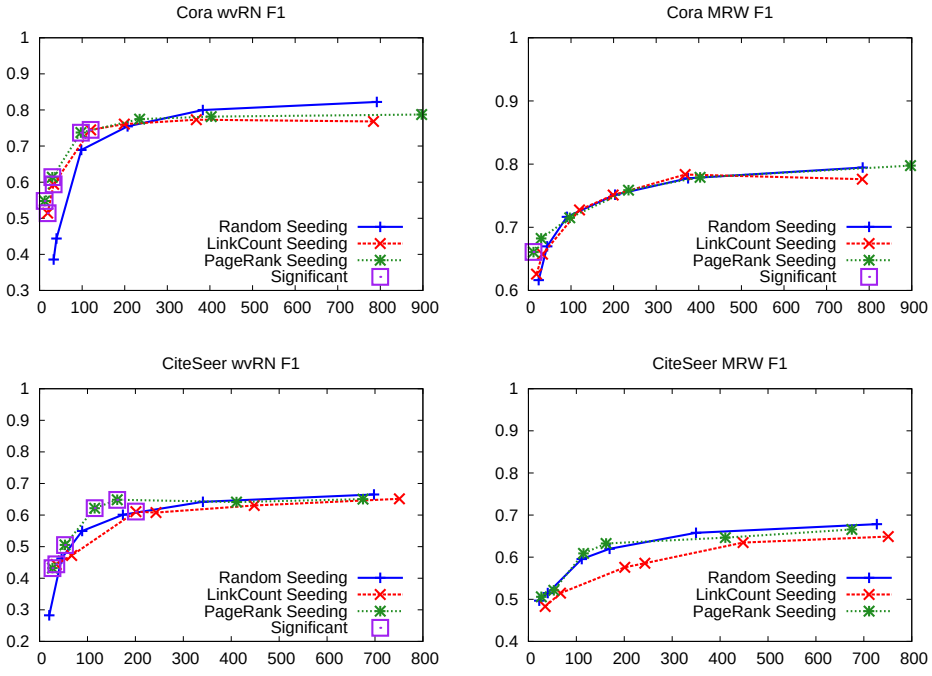


Figure 4: Citation datasets results the seeding method. The x-axis indicates number of labeled instances and y-axis indicates labeling macro-averaged F1 score. Square block around a point indicates statistical significance with $p < 0.05$.

6 Why Do Random Graph Walk-based Methods Work Better?

At this point, we like to ask a very important question — Why does MRW work better than wvRN (or the Gaussian fields classifier)? And why is the difference more pronounced when given random seeds? Actually, the above result showing that *using high authority seeds greatly boosts the classification performance of wvRN* offers an important clue why MultiRankWalk (or other semi-supervised learning methods based on random graph walk) may work better.

If we consider high authority instances as having more classification power (either due to how the graph is naturally formed or due to them being more"connected" to other instances), we take advantage of that power when we use them in wvRN as seeds and that power is lost when seeds are chosen randomly. To see why, we take a look at the Equation 6.2 that define the class probability in wvRN [13] and Equation 6.3 that is the harmonic property of the unlabeled points [22]:

$$P(x_i = c|N_i) = \frac{1}{Z} \sum_{v_j \in N_i} w_{i,j}.P(x_j = c|N_j) \tag{6.2}$$

$$f(j) = \frac{1}{d_j} \sum_{i \; j} w_{i,j}.f(i) \tag{6.3}$$

In both cases, either probability $P(x_i)$ or the function $f(j)$, the maximum value is 1, which is the constant value for all labeled instances, and any unlabeled instance cannot have a value more than a labeled instance. This property, or limitation, prevents any instance, regardless of the graph structure, to have more "influence" over the graph (or the Gaussian field) than a labeled seed. A particular unlabeled instance could in fact be well-connected to several labeled seeds of the same class and many unlabeled instance, and therefore should have more influence over the network. When there are many labels this constraint is probably not important but when the number of labeled instances is small and distributed randomly, the function over the Gaussian field may be bumpy rather than smooth.

Graph walk methods, on the other hand, do not have this constraint; a unlabeled instance could have much more influence on the graph (i.e., having a higher per-class rank than a seed instance), which exploits more fully the power of the unlabeled data. A toy example shown in Figure 8 illustrates this idea. The top graph shows nodes before running any classifiers; an **S** indicates labeled seed nodes and the color indicates class prediction. The middle graph shows node labels predicted by wvRN and the bottom graph shows node labels predicted by MRW. The relative sizes of nodes in the graphs indicate how strongly the algorithm believes the node labels to be of the color shown. Besides having a more reasonable prediction, the sizes of nodes in the bottom graph show that graph walk predictions results in a smoother propagation graph with the center of the "clusters" having the highest confidence, and higher confidence in turn means stronger propagation influence. In addition, it shows that wvRN is more sensitive to the location of the labeled seed nodes; in this case, having a seed node near the fringe of the "cluster" resulted in an incorrect prediction.

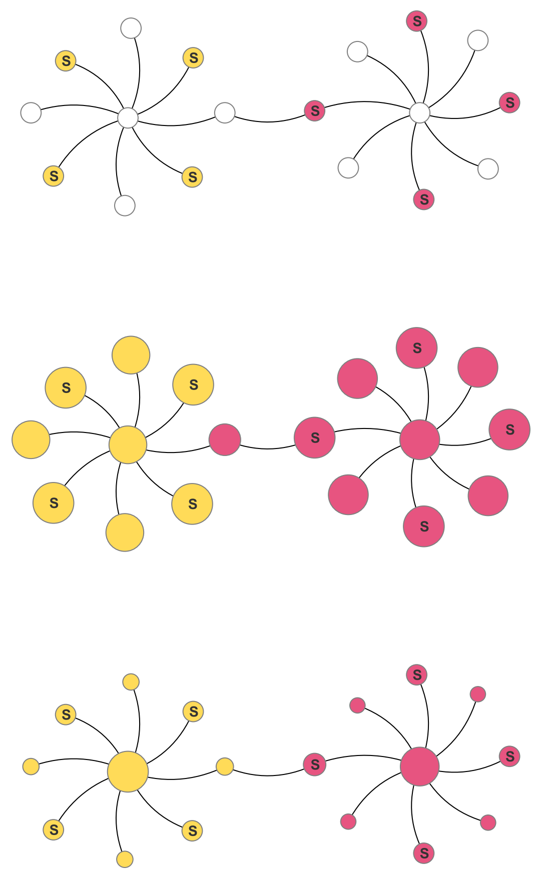


Figure 8: Top: Labeled and unlabeled nodes before prediction. Middle: Labels predicted by wvRN. Bottom: Labels predicted by MRW. The relative sizes of nodes in the graphs indicate how strongly the algorithm believes the node labels to be of the color shown and how strongly one node will influence its neighbors.

Figure 9 shows what happens when the seed nodes are moved to the center of the clusters. Both methods now give reasonable label predications. Looking at the

relative sizes of the nodes, we see that while the resulting propagation graph of wvRN is rather different from Figure 8, the propagation graph of MRW seems to be the same. This matches the experimental results where high authoritative are used as seeds. This suggests that MRW is less sensitive to the exact placement of seed nodes and still able to produce a smooth propagation graph where more authoritative nodes tend to have more influence.

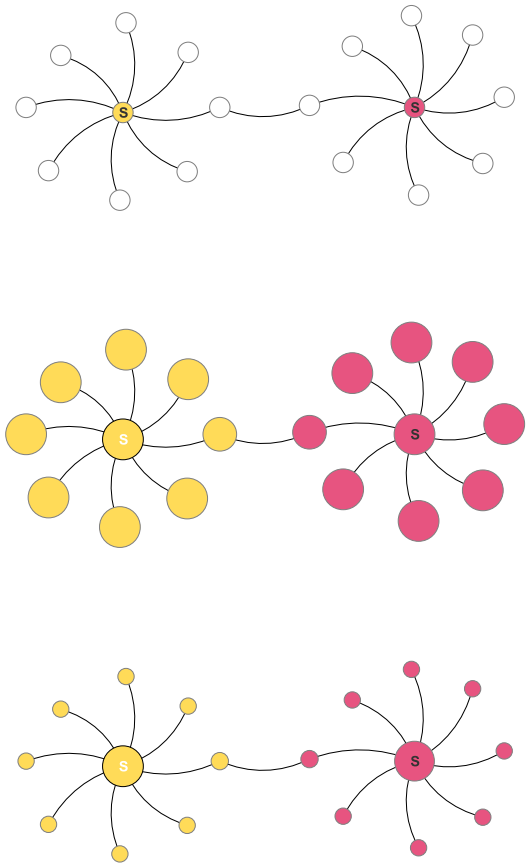


Figure 9: Top: Labeled and unlabeled nodes before prediction. Middle: Labels predicted by wvRN. Bottom: Labels predicted by MRW. The relative sizes of nodes in the graphs indicate how strongly the algorithm believes the node labels to be of the color shown and how strongly one node will influence its neighbors.

7 Scalability

The best seed preference algorithm is based on PageRank, so the run time is linear to the number of edges in the graph and converges fairly quickly even when applied to large graphs [15]. The proposed algorithm is based on random graph walk with restart, and the run time is also linear to the number of edges in the graph; the core algorithm itself has been well-studied and several performance-enhancing methods have been proposed to minimized the amount of storage and time required such as the one found in [18].

8 Conclusions

We proposed MultiRankWalk, a semi-supervised learning method as a simple yet intuitive representative of a class of semi-supervised learning methods based on random graph walks, and show it to significantly outperform other semi-supervised and supervised learning methods when only a few labeled instances are given on five network datasets.

We also show that using high authority labeled instances dramatically reduce the amount of labels required to achieve high classification performance, which sheds light on why random graph walk-based methods have an advantage over methods such as Gaussian fields classifier when the size of training data is small.

Due to this advantage, its simplicity, and classification accuracy, we highly recommend MultiRankWalk as a strong baseline for future graph-based semi-supervised learning experiments.

References

[1] Nielsen Buzzmetrics, www.nielsenbuzzmetrics.com.

[2] L. Adamic and N. Glance. The political blogosphere and the 2004 u.s. election: Divided they blog. In *Proceedings of the WWW-2005 Workshop on the Weblogging Ecosystem*, 2005.

[3] W. W. Cohen. Improving a page classifier with anchor extraction and link analysis. In *Advances in Neural Information Processing Systems 15*, 2002.

[4] M. Craven, D. DiPasquo, D. Freitag, A. K. McCallum, T. M. Mitchell, K. Nigam, and S. Slattery. Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence*, 118(1/2):69–113, 2000.

[5] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

[6] Z. Gyongyi, H. Garcia-Molina, and J. Pedersen. Web content categorization using link information. Technical report, Stanford University, 2006.

[7] T. Haveliwala, S. Kamvar, and G. Jeh. An analytical comparison of approaches to personalizing pagerank. Technical report, Stanford University, 2003.

[8] J. He, J. Carbonell, and Y. Liu. Graph-based semi-supervised learning as a generative model. In *International Joint Conferences on Artificial Intelligence*, 2007.

[9] A. Kale, A. Karandikar, P. Kolari, A. Java, T. Finin, and A. Joshi. Modeling trust and influence in the blogosphere using link polarity. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*, 2007.

[10] Z. Kou and W. W. Cohen. Stacked graphical models for efficient inference in markov random fields. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, 2007.

[11] Q. Lu and L. Getoor. Link-based classification. In *Proceedings of the 20th International Conference on Machine Learning*, 2003.

[12] S. A. Macskassy. Improving learning in networked data by combining explicit and mined links. In *The Twenty-Second Conference on Artificial Intelligence*, 2007.

[13] S. A. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *The Journal of Machine Learning Research*, 8.

[14] A. Y. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, 2002.

[15] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[16] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[17] M. Szummer and T. Jaakkola. Partially labeled classification with Markov random walks. In *Advances in Neural Information Processing Systems 14*, 2001.

[18] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast randomwalk with restart and its applications. In *Proceedings of the 2006 IEEE International Conference on Data Mining (ICDM 2006)*, 2006.

[19] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2001.

[20] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, 2004.

[21] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

[22] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *The 20th International Conference on Machine Learning*, 2003.

[23] X. Zhu, J. Lafferty, and Z. Ghahramani. Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions. In *ICML 2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003.