Empirical comparison of hard and soft label propagation for relational classification

Conference Paper · June 2007

DOI: 10.1007/978-3-540-78469-2_13 · Source: dx.doi.org

2 authors:

Aram Galstyan
University of Southern California
268 PUBLICATIONS **9,918** CITATIONS

SEE PROFILE

Paul R. Cohen
The University of Arizona
393 PUBLICATIONS **7,795** CITATIONS

SEE PROFILE

Empirical Comparison of "Hard" and "Soft" Label Propagation for Relational Classification

Aram Galstyan and Paul R. Cohen

USC Information Sciences Institute
Center for Research on Unexpected Events (CRUE)
Marina del Rey, CA, USA
{galstyan,cohen@isi.edu}

Abstract. In this paper we differentiate between *hard* and *soft* label propagation for classification of relational (networked) data. The latter method assigns probabilities or class-membership scores to data instances, then propagates these scores throughout the networked data, whereas the former works by explicitly propagating class labels at each iteration. We present a comparative empirical study of these methods applied to a relational binary classification task, and evaluate two approaches on both synthetic and real–world relational data. Our results indicate that while neither approach dominates the other over the entire range of input data parameters, there are some interesting and non–trivial tradeoffs between them.

1 Introduction

Many relational classification algorithms work by iteratively propagating information through relational graphs. The main idea behind iterative approaches is that "earlier" inferences or prior knowledge about data instances can be used to make "later" inferences about related entities. Examples include relaxation labeling for hypertext categorization[1], belief propagation for probabilistic relational models [2], relevance propagation models for information retrieval on the web [3], iterative label propagation [4, 5], relational neighbor classifiers [6–8].

While there are various ways to propagate information through relational graphs, in this paper we differentiate between two general approaches: In the first approach, hard class label assignments are made at each iteration step. In this paper we call this approach label propagation[1] (LP). The second approach, which we call Score Propagation (SP), propagates soft labels such as class membership scores or probabilities. To illustrate the difference between these approaches, assume that we want to find fraudulent transaction given a relational graph of transactions (such as in Figure 1) and some known fraudulent nodes. For each transaction we could estimate the probability of it being fraudulent using information about the nodes it connects and their neighbors. The SP algorithm propagates these probabilities throughout the system, and then makes a final inference by projecting the probabilities onto class labels. The LP algorithm, on the other hand, estimates these probabilities at the first step, finds the entities with the

[1] We note that sometimes the term "label propagation" is also used to describe soft–label propagation.

highest probability of being fraudulent, labels them as fraudulent, and then iterates this procedure.

We would like to emphasize that despite the term "hard label propagation", in the present paper we focus on comparing two algorithms with respect to their accuracy of ranking rather than explicit classification. For the ranking problem, the difference between two approaches can be explained as follows: The SP algorithm is analogous to a diffusion-like process on a network, where initially labeled nodes act as heat sources, and the rank of a node is determined by its *temperature*. The LP algorithm, on the other hand, is similar to a discrete epidemic model, where, starting from initially *infected* nodes, the epidemic spreads to other nodes, and a node's rank depends on how early in the epidemic process it was infected.

Intuitively, one could think that the LP algorithm described above would not perform as well as soft label propagation, since it makes hard "commitments" that cannot be undone later when more information is propagated through the network. The main finding of this paper is that this not always the case. We present results of extensive experiments for a simple binary classification task, using both synthetic and real–world data. For synthetic data we empirically evaluate the difference in performance of both algorithms for a wide range of input parameters. We find that LP is usually a better choice if the overlap between the classes is not strong. More interestingly, we found that even when the performance of two algorithms are similar in terms of their AUC (area under the curve) score, *two algorithms might have significantly different accuracy for an allowed false positive rate*, e.g., they have different ROC (Receiver–Operator Characteristics) curves. The other important observation is that for certain data parameters the LP algorithm is much more robust to the presence of noise in the initial class label assignment. In other words, *our results suggests that for noisy data, propagating hard labels instead of scores might be a better choice*.

In addition to our experiments on synthetic data, we tested both algorithms on CoRA data–set of hierarchically categorized computer science papers. We constructed a separate classification problem for each CoRA sub–topic in Machine Learning category. Despite certain differences between our results for CoRA and synthetic data, we observed that hard label propagation scheme is indeed more robust with respect to noise, for the majority of the topics considered. Our CoRA experiments also reproduced the different ROC behavior for certain topics, although this difference was not as large as in the case of synthetic data.

The rest of this paper is organized as follows: in the next section we describe the binary classification problem and synthetic data used in our experiments. We introduce hard and soft label propagation algorithms in Section 3. Section 4 describes related work. The results of experiments on synthetic and CoRA data are presented in Sections 5 and 6 respectively. Concluding remarks are made in section 7.

2 Problem Settings

Most relational classification techniques rely on both intrinsic and relational attributes of the data for making inferences. For instance, if the task is to classify scientific papers into topics, both intrinsic features (e.g., frequency of certain keywords) and relational

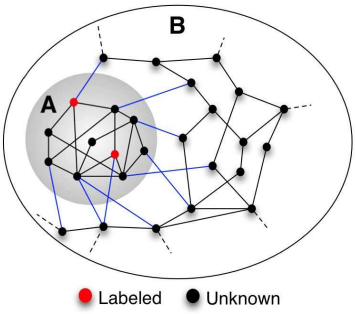

Fig. 1. Schematic representation of networked data.

attributes (e.g., common author, references, etc.) may be used. In this paper we are mainly interested in relational aspect of classification, so we ignore intrinsic attributes of data instances and instead examine the effects of relational structures on classification accuracy. Thus, the data is represented by an undirected graph, where nodes correspond to data instances and edges represent relationships between them. Now we state the classification problem that we are interested in. Assume a relational graph as schematically illustrated in Figure 1. In the classification problem one wants to find the set $\mathcal{A}$ of nodes that belong to class A (the shaded region), given the relational graph and a small subset $\mathcal{A}^0 \in \mathcal{A}$ of labeled A instances. In the ranking problem addressed in this paper, we are merely interested in ranking the nodes according to their similarity to class A. We denote the nodes not in A as class B, and the corresponding set as $\mathcal{B}$. In general, class B itself might comprise other classes that will be reflected in the topology of the network. This is the case for the CoRA data studied in Section 6. For the synthetic data, however, we will assume a homogenous structure for each class. Specifically, within each class, we randomly distribute links between pairs of nodes with probability $p_{in}^{a,b}$ so that the relational structures within the classes are characterized by Erdos–Renyi graphs $G(N_A; p_{in}^a)$ and $G(N_B; p_{in}^b)$[2]. N_A and N_B are the number of nodes in respective classes. Then we randomly establish links across the classes (blue edges in Fig. 1), by assigning a probability p_{out} to each of $N_A N_B$ possible links. The average number of links per node (connectivities) within and across the classes are given by $z_{aa} = p_{in}^a N_A$, $z_{bb} = p_{in}^b N_B$, $z_{ab} = p_{out} N_B$ and $z_{ba} = p_{out} N_A$. If the sizes of two classes are not equal then $z_{ab} \neq z_{ba}$.

Note that our construction of the synthetic relational graph enforces the *homophily condition* which means that better-connected nodes are likely to be in the same class. Hence, we should expect the difficulty of the classification task to be strongly affected by the ratio of connectivities within and across the classes. We will use $z_{ab}/z_{aa} \equiv z_{out}/z_{in}$ to characterize the degree of homophily. A small value of this ratio means that the classes are well–separated (strong homophily) so most classification algorithms should do a good job of assigning correct class labels. For large values of z_{out}/z_{in},

[2] Erdos–Renyi graph G(N; p) is constructed by independently linking each pair of N nodes with probability p.

on the other hand, the difference between link patterns within and across the classes decreases, making it more difficult to classify nodes correctly. We examine the effects of class overlap in the experiments described in Section 5.

3 Algorithms

The score propagation mechanism employed in this paper is very similar to suspicion scoring model of Macskassy and Provost [9], as well as to relevance propagation techniques from information retrieval literature [3, 10]. The label propagation algorithm, on the other hand, can be viewed a discrete (binary) analogue of the score propagation scheme. Below we describe both approaches in more details.

3.1 Score Propagation

By score propagation we mean a type of iterative procedure that propagates continuous–valued class membership scores from labeled class instances to unlabeled ones. In our example the initially labeled set contains only nodes of type A. Hence, we associate a single score with each node that describes its relative likelihood of being in class A. These scores are updated iteratively, allowing the influence of labeled nodes to spread throughout the data. The main assumption behind this scheme is that nodes with higher scores are more likely to be member of the sought class.

There are many possible ways to implement the a propagation mechanism. Here we employ a scheme described by the following equation:

$$s_i^{t+1} = s_i^0 + \alpha_i s_i^t + \beta \sum_j W_{ij} s_j^t \tag{1}$$

Here s_i^0 is a static contribution that might depend on node's intrinsic attributes, α_i and β_i are parameters of the model, and $W_{ij} = 1$ if nodes i and j are connected and $W_{ij} = 0$ otherwise. For instance, in the relevance propagation models in information retrieval, s_i^0 is the content-based self-relevance score of a node, $\alpha_i = const < 1$, and $\beta_i = (1 - \alpha_i)/z_i$, where z_i is the connectivity of node i. In the suspicion scoring model of Ref. [9], $s_i^0 = 0$, $\alpha_i = \alpha$ for all i, and $\beta = (1 - \alpha)/\sum_{i,j} W_{ij}$.

Our experiments with variants of SP schemes suggest that they all behave in qualitatively similar ways. For this paper, we report results for a simple parameter-free version obtained by setting $s_i^0 = \alpha_i = 0$, and $\beta_i = 1/z_i$. The resulting updating scheme is:

$$s_i^{t+1} = \frac{1}{z_i} \sum_j W_{ij} s_j^t \tag{2}$$

In other words, at each time step the class membership score of a node is set to the average scores of its neighbors at the previous step. We note the resemblance of this model to the random walk model of Ref. [11].

Initially, the scores of labeled A nodes are fixed to 1, while the rest of the nodes are assigned score 0. Because of clamping, the former nodes act as diffusion sources, so that the average score in the system increases with time and in fact converges to 1.

Therefore, we stop the iteration after the average score exceeds some threshold, chosen to be 0.9 in the experiments reported below. We observed that the final ranking of the nodes according is not sensitive to the choice of this threshold.

3.2 Label Propagation

For label propagation (LP) we developed a simple mechanism that is in some sense the discrete (binary) analogue of the SP scheme. Let us assign binary state variables $\sigma_i = \{0, 1\}$ to all nodes so that $\sigma_i = 1$ (or $\sigma_i = 0$) means that the i–th node is labeled as type A (or is unlabeled). At each step of iteration, for each unlabeled node, we calculated the fraction of the labeled nodes among its neighbors, $\omega_i^t = \sum_j W_{ij}\sigma_j^t/z_i$, and then label the nodes for which the fraction is the highest. This procedure is then repeated for T_{max} steps.

The label propagation algorithm above can be viewed as a combination of the scoring propagation scheme from the previous section and a nonlinear (step-function-like) transformation applied after each iteration. This nonlinear transformation constitutes a simple inference process where the class-membership scores of a subset of nodes are projected into class labels. This happens at every inference step. Indeed, assume that starting from the initially given labels, we iterate the SP scheme of Equation 2 once. Then, obviously, $s_i^1 = \omega_i^1$. That is, the nodes with maximum fractions of labeled nodes among neighbors also have the highest score. The step-like transformation then assigns score 1 to all the nodes sharing the maximum score, and sets the score of the remaining nodes to zero, thus acting as a filter.

While ranking nodes in the SP scheme is straightforward, we need a different ranking mechanism for the LP scheme. Note that the only parameter of the LP classification scheme is the iteration length T_{max}. In particular, by choosing different T_{max} one effectively controls the number of labeled instances. Hence, setting T_{max} is in a sense analogous to setting a classification threshold for the SP mechanism. This suggests the following natural criterion for ranking: Rank the nodes according to the iteration time step when they were labeled as type A, so that a node that is classified earlier in the iteration has a lower rank (i.e., is more likely to belong to the class A). The justification of this approach is again based on the homophily condition: nodes that are similar to the initially labeled nodes will tend to be better connected with them, hence they will be labeled earlier in the iteration.

4 Related Work

Before presenting our experimental results, we would like to clarify the connection of the models in section 3 with existing work. The score propagation model Equation 2 is a special case of the suspicion scoring model of Macskassy and Provost [9]. One subtle difference is that Ref. [9] uses annealing to guarantee convergence, by decreasing α with time. Another aspect of the work in [9] is adaptive data access based on the iterative runs of the scoring scheme. Specifically, after a first run of the SP scheme, they choose the top K nodes and query them against a secondary database and augment the network with new links. Then they run the SP scheme again to generate new rankings. Since

in our model the relational graph is given initially, we do not perform iterations over many SP schemes. We note, however, that our LP algorithm is analogous of performing multiple iterations over the score propagation scheme where each SP run includes only one iteration of Equation 2.

Recently there has been a growing interest in the web-based information retrieval community in using both link and content information for web queries [12]. The SP model is strongly related to relevance propagation schemes from web-based information retrieval [3, 10]. One of the differences is that our model does not have the self–relevance term that describes a node's content. Also, the graph in our model is undirected, while for web mining the link directionality plays an important role (see also [13]).

The classification problem considered here is related to semi-supervised learning with partially labeled data. Recently, several algorithms that combine both labeled and unlabeled data have been suggested [11, 14, 15]. Remarkably, these approaches too are based on the homophily assumption that nearby data points are likely to belong to the same class. Given a dataset with partially labeled examples, [15] construct a fully connected graph so that the weight of the edge between two points depends on the distance $d(x_1, x_2)$. They then suggests a "soft" label propagation scheme where the information about the labeled nodes is propagated throughout the constructed graph. Because of their problem formulation, they were able to avoid the actual propagation step and instead solve a linear system of equation. Despite obvious similarities, there are also important differences with the model considered here. First, the scores in our model are not interpretable as probabilities. Also, the algorithm in Ref [15] works only if there are initially labeled data points from both clusters (for binary classification), while in our case we do not have that constraint.

5 Experiments with Synthetic Data

We evaluated the performance of the SP and LP algorithms using ROC curve analysis, and particularly, AUC (Area Under the ROC Curve) scores. In our experiments with synthetic data, we used equal class sizes, $N_A = N_B = 500$ for one of the experiments, and skewed class distribution with $N_A = 200$ and $N_B = 2000$ in all the others. We run 100 trials for each choice of parameters, and calculated both the average and the standard deviation of AUC score over the trials.

5.1 Class Overlap

In the first set of experiments, we examine the effect of class overlap on classification accuracy. As we already mentioned, the class overlap can be measured by the ratio z_{out}/z_{in}. In Figure 2 we plot the AUC score against the ratio z_{out}/z_{in} for three different values of z_{in}. The top panel shows the results for equal class sizes, $N_A = N_B = 500$, with the number of initially labeled instances $N_A^0 = 100$, e.g., 20% of all A nodes. Starting from near-perfect AUC scores at the ratio 0.1 for $z_{in} = 5$, the accuracy of both SP and LP degrades gradually while increasing the ratio z_{out}/z_{in}, and, as we expected, falls to 0.5 for $z_{in} \approx z_{out}$. We also note that there is a crossover region in the

performances of both algorithms: at $z_{out}/z_{in} = 0.1$, LP attains slightly higher AUC score than SP, while for $z_{out}/z_{in} \geq 0.5$ the SP algorithm peforms better. This pattern is amplified by larger within–class connectivity. Indeed, for $z_{in} = 20$ both algorithms attain perfect AUC score for ratios up to 0.3, and then, for $z_{out}/z_{in} > 0.3$, LP clearly outperforms SP up until the crossover point at 0.7, with the difference in the AUC scores as high as 0.1 at certain points. More interestingly, the crossover point where SP starts to perform equally and then better shifts right with increasing within-class link density. This suggests that for sufficiently dense graphs, the LP algorithm is a better choice if the class overlap is not very large. For sparse graphs and relatively large overlap, however, SP performs better.

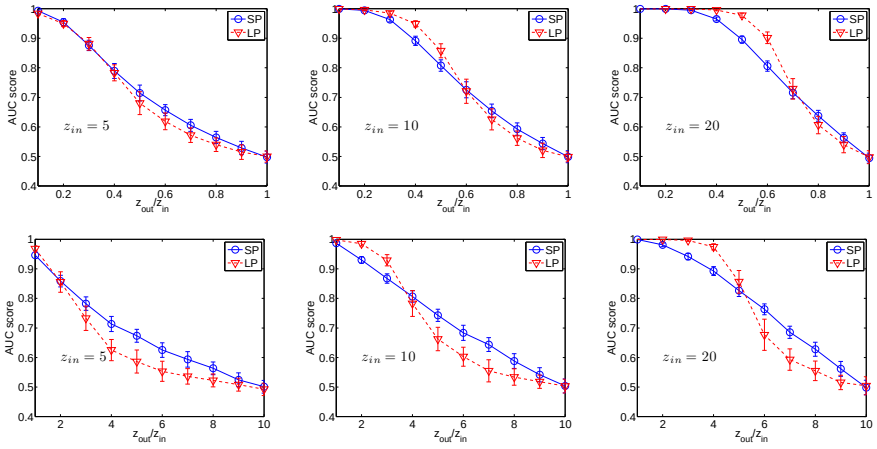

Fig. 2. AUC score vs the ratio z_{out}/z_{in} for different values of z_{in}. The top and bottom panels are for equal and skewed class distributions respectively.

A similar picture holds in the presence of class skew (bottom panel in Fig. 2). The number of nodes in each class are $N_A = 200$ and $N_B = 2000$, with again 20% of A nodes initially labeled (i.e., $N_A^0 = 40$). The only difference from the equal class size scenario is that the ratio at which the performance of both algorithms falls to a random level is now shifted towards higher values of z_{out} (note that the horizontal axis ranges from 1 to 10). The reason for this is that for a given z_{out}, the average number of type A neighbors for type B nodes is $z_{ba} = z_{out}N_A/N_B$. One should expect the ranking to be random at the overlap level when a B node has roughly same number of A neighbors as A nodes, themselves. Hence, we can estimate this ratio as $z_{out}/z_{in} \sim N_B/N_A$. For the class skew of 10 one gets $z_{out}/z_{in} \sim 10$, which agrees well with the experiment.

5.2 ROC analysis

We now describe differences in the performance of both algorithms observed in the ROC curves for $z_{in} = 5$ and three different choices of class overlap: $z_{out} = \{5, 10, 15\}$.

For $z_{out} = 5$ the LP algorithm achieves a slightly better AUC score than the SP. For $z_{out} = 10$ both algorithms have the same AUC score (within the standard deviation). And finally, for $z_{out} = 15$ SP has a better AUC score (see the bottom panel inFig. 2). In the experiments, we used bins of size 0.01 for the false positive rates FP. For each bin we calculated the average and standard deviation of the corresponding true positive rates TP. The results are shown in Fig. 3.

Let us first discuss the the case $z_{out} = 5$. The corresponding AUC scores are 0.95 ± 0.01 for SP and 0.97 ± 0.01 for LP. What is remarkable, however, is that despite this tiny difference, the two classifiers are quite distinct for small false positive rates. In other words, the difference in AUC score is not distributed equally over the whole ROC plane. Instead, the main difference is for FP rate between 0 and 0.1. For false positive rates of larger than 0.3, on the other hand, SP achieves marginally better true positive rates. This observation suggests that if the cost of false positives are high, then LP is a superior choice for small class overlap. This can be especially important in the case of highly skewed class distributions, where even tiny false positive rates translate into large numbers of falsely classified instances. The inset shows the difference between true positive rates $\Delta TP = TP_{LP} - TP_{SP}$ at a fixed false positive rate. Along with each point, we plot bars that are two standard deviations wide and centered around the mean. Clearly, for a small interval around $FP = 0.05$, this difference is positive and statistically significant, and achieves a value as high as ~ 0.3.

A somewhat similar, although less dramatic, effect holds for $z_{out} = 10$. Note that the AUC scores of both algorithms are indistinguishable. In this case, LP achieves better true positive rates in the interval $FP \in [0; 0.2]$, while SP performs better on the rest of the axis. The difference between them is not as pronounced as it is with smaller class overlap (note also the higher standard deviations). Finally, for $z_{out} = 15$ the SP algorithm matches the performance of LP for small positive rates, and outperforms the latter over the rest of the FP axis. This again suggests that for relatively large class overlap SP is a better choice. Followup experiments revealed that the observed differences in ROC curves, especially for small false positive rates, persist for wide ranges of parameter choices as long as the overlap between the classes is not very large. Moreover, the difference becomes more dramatic for larger within-class connectivities, z_{in}. For some parameters this difference was as high as 0.5 for small FP rates.

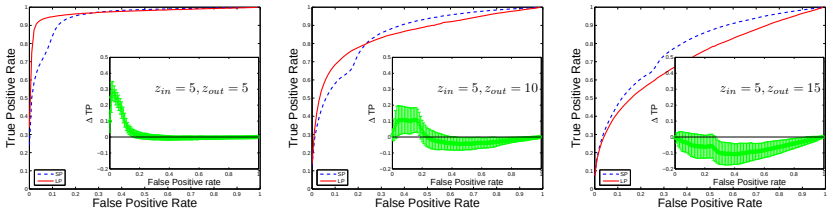

Fig. 3. ROC curves for different connectivities.

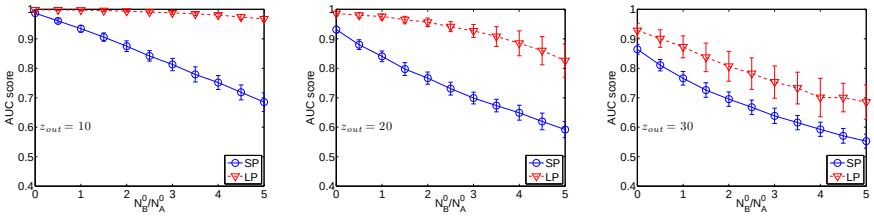

Fig. 4. The AUC score plotted against the ratio N_B^0/N_A^0. The within–class connectivity is $z_{in} = 10$.

5.3 Effect of noise

Next, we study how the classification accuracy deteriorates in the presence of noise, which was introduced by randomly and uniformly choosing N_B^0 nodes from B and mislabeling them as type A in the initial data set. In the experiments, we set the number of initially labeled A nodes to $N_A^0 = 40$, and studied how the AUC score changes as we increased the number of mislabeled nodes, N_B^0. The results are presented in Fig. 4 where we plot the AUC score against the ratio N_B^0/N_A^0 (for three different values of the class overlap). Remarkably, for small class overlap, $z_{out} = 10$, the noise has a distinctly different effect on SP and LP. The LP algorithm seems to be very resilient to the noise and has an AUC score close to ~ 0.97 even when the number of mislabeled nodes is $N_B^0 = 200$, or five times the number of correctly labeled nodes. The performance of the SP algorithm, on the other hand, deteriorates steadily starting from moderate values of noise and attains an AUC score of only 0.68 for $N_B^0 = 200$. A similar, although weaker, effect is observed for moderate overlap $z_{out} = 20$. The AUC score of the SP algorithm decreases at a nearly linear rate, while for the LP scheme the decrease is much slower. Finally, for $z_{out} = 30$ the noise seems to affect the performance of both algorithms in very similar ways.

6 Experiments with CoRA Data

The assumption that the relational structure is described by coupled Erdos-Renyi graphs might not be appropriate for real world datasets. Hence, it is important to find out whether the results described in the previous sections hold for more realistic data. In this section we present the results of our experiments on CoRA data–set of hierarchically categorized computer science research papers [16]. We focus on the papers in the Machine Learning category, which contains seven different subtopics: “Case-Based”, “Genetic Algorithms”, “Probabilistic Methods”, “Neural Networks”, “Reinforcement Learning”, “Rule Learning” , and “Theory”. Two papers were linked together by using common author (or authors) and citation. After pruning out the isolated papers from the data–set, we were left with 4025 unique titles. In our experiments we mapped the multi–class problem onto a binary classification problem for each individual topic.

Generally speaking, the results obtained for CoRA data were somewhat different from results for the synthetic data. Specifically, we found that the ranking accuracies

were lower than one would expect for a random Erdos–Renyi topology with corresponding connectivities, especially for the LP algorithm. We believe that this is due to the fact that the CoRA graph has a much more skewed degree distribution compared to the exponential distribution of Erdos–Renyi graphs (indeed, we established that the performances of both algorithms improve if we purge nodes with very high and very low connectivities from the graph). We also found that in contrast to the synthetic data, the SP algorithm was usually better than LP in case where there was no noise in the initial label assignment.

Despite these differences, however, we established that our main results for the synthetic data held for some of the CoRA topics. In particular, we observed that for four out of seven topics the LP algorithm is indeed less sensitive to noise. This is shown in Figure 5 where we plot the AUC score vs the fraction of mislabeled nodes for six of the seven topics. For the "Genetic Algorithms", "Reinforcement Learning", "Rule Learning" , and "Theory" topics, the decrease in accuracy for the LP algorithm is smaller than for the SP algorithm, although the difference is not as dramatic as for the synthetic data. For two other topics, "Case-Based" and "Probabilistic Methods", as well as for the "Neural Networks" topic not shown here, the response of both algorithms to noise did not differ much.

Further, in Figure 6 we show the ROC curves for the same topics. Again, for some of the topics the observed picture is qualitatively very similar to that presented in Figure 3 for synthetic data. Namely, although the overall accuracy of both classifiers (e.g., AUC scores) are very close, their ROC curves are different, with LP algorithm achieving better accuracy for small false positive rates. This is especially evident for the "Reinforcement Learning" subtopic for which the (average) maximum difference is close to 0.18. Note also that for "Case-Based" and "Probabilistic Methods" topics SP outperforms LP for the whole ROC plane (this is also true for the topic "Neural Networks").

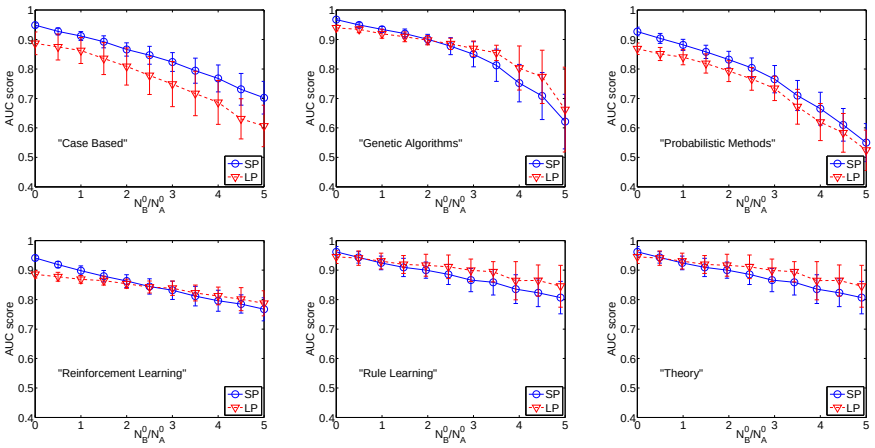

Fig. 5. The AUC score plotted against the ratio N_B^0/N_A^0 for different CoRA topics.

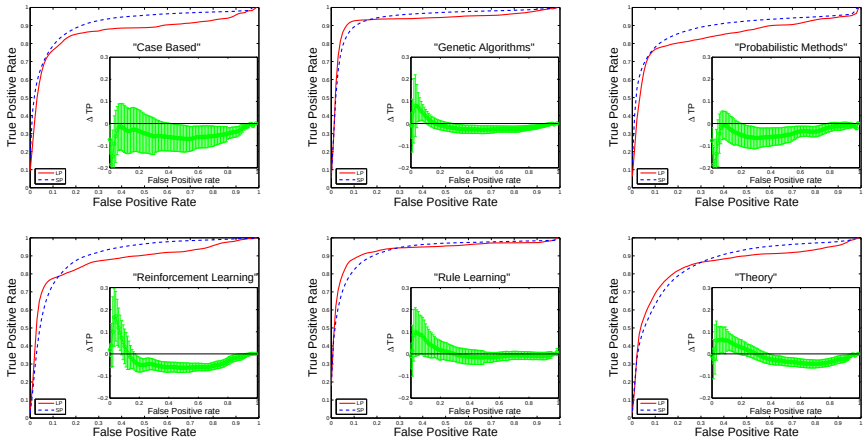

Fig. 6. ROC curve for different CoRA topics.

7 Discussion and Future Work

In this paper we have presented empirical comparison of *hard* and *soft* label propagation techniques for binary relational classification. Our results suggest that for sufficiently strong homophily of the linked data, both methods achieve a remarkably good ranking accuracy. We also found that, while neither of the approaches dominates over the entire range of input parameters, there are some important differences that should be taken into account for deciding which one is better suited for a particular problem.

One of the main findings of this paper is that even when two algorithms achieve the same accuracy of ranking (as characterized by their AUC scores), the behavior of the family of classifiers based on them can be drastically different. Specifically, we found that for small values of allowed false positive rates, LP usually achieves higher true positive rates. In fact, for data with small class overlap, the observed difference was quite dramatic. The SP algorithm, on the other hand, achieves higher true positive rates for larger allowed false positive rate. This suggests that SP might be a better choice only when the cost of false negatives strongly outweighs the cost of false positives. This difference will be especially important in the case of highly skewed class distributions, where even tiny false positive rates translate into a large number of falsely classified instances.

The other important finding of this paper is the different behavior of the two propagation schemes in the presence of noise. Our experiments with synthetic data, as well as for some of the CoRA topics, suggest that LP algorithm is less more robust to mislabeled data instances. Thus, propagating hard labels instead of scores might be a better choice when the prior information is noisy. We believe that this is an interesting observation that warrants a further examination, both analytically and empirically.

We also note the algorithms have different computational complexities. Indeed, the worst case time–complexity of the LP algorithm scales linearly with the number of data instances, as it might require N iterations to rank N instances. This correspond to the

case when only one node is labeled at each iteration step. The SP algorithm, on the other hand, scales much better with the data size. In fact, our experiments show that the relative ranking almost saturates once the influence propagates from the seed nodes to the rest of the nodes, which happens after much shorter time scale (order of $\sim \log N$). This difference can be very important for very large scale data.

Many relational classification techniques rely on information propagation over graphs. However, there are not many systematic studies that examine the role of the graph structure on the propagation dynamics. In this paper we have addressed this problem for fairly simple propagation dynamics and graph topology. We believe it would be worthwhile to perform similar studies for more sophisticated classification schemes, and extend the empirical framework presented here to more complex relational domains. Currently, evaluations of various relational learning algorithms are limited to a handful of real world datasets. While it is important to perform well on real world data, we believe that evaluating an algorithm through a controlled set of experiments on synthetic data will help to better understand its strengths and weaknesses.

References

1. Chakrabarti, S., Dom, B.E., Indyk, P.: Enhanced hypertext categorization using hyperlinks. In Haas, L.M., Tiwary, A., eds.: Proceedings of SIGMOD-98, ACM International Conference on Management of Data, Seattle, US, ACM Press, New York, US (1998) 307–318
2. Friedman, N., Getoor, L., Koller, D., Pfeffer, A.: Learning probabilistic relational models. In: Proceedings of the IJCAI-1999. (1999) 1300–1309
3. Qin, T., Liu, T.Y., Zhang, X.D., Chen, Z., Ma, W.Y.: A study of relevance propagation for web search. In: SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (2005) 408–415
4. Galstyan, A., Cohen, P.R.: Inferring useful heuristics from the dynamics of iterative relational classifiers. In: Proceedings of IJCAI-05, 19th International Joint Conference on Artificial Intelligence. (2005)
5. Galstyan, A., Cohen, P.R.: Relational classification through three–state epidemic dynamics. In: Proceedings of the 9th International Conference on Information Fusion, Florence, Italy (2006)
6. Macskassy, S., Provost, F.: A simple relational classifier. In: Proceeding of the Workshop on Multi-Relational Data Mining in conjunction with KDD-2003 (MRDM-2003), Washington, DC (2003)
7. Macskassy, S., Provost, F.: Classification in networked data: A toolkit and a univariate case study. Working paper CeDER-04-08, Stern School of Business, New York University (2004)
8. Macskassy, S., Provost, F.: Netkit-srl: A toolkit for network learning and inference. In: Proceeding of the NAACSOS Conference. (2005)
9. Macskassy, S., Provost, F.: Suspicion scoring based on guilt-by-association, collective inference, and focused data access. In: Proceeding of the International Conference on Intelligence Analysis, McLean, VA (2005)
10. Shakery, A., Zhai, C.: Relevance propagation for topic distillation uiuc trec 2003 web track experiments. In: TREC. (2003) 673–677
11. Szummer, M., Jaakkola, T.: Partially labeled classification with markov random walks. In: Advances in Neural Information Processing Systems. Volume 14. (2001)

12. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project (1998)
13. Gyongyi, Z., Garcia-Molina, H., Pedersen, J.: Combating web spam with trustrank. In: In Proceedings of the 30th VLDB Conference. (2004)
14. Tishby, N., Slonim, N.: Data clustering by markovian relaxation and the information bottleneck method. In: NIPS. (2000) 640–646
15. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Technical report, Carnegie Mellon University (2002)
16. McCallum, A., Nigam, K., Rennie, J., Seymore, K.: Automating the construction of internet portals with machine learning. Information Retrieval Journal **3** (2000) 127–163