Empirical Evaluation of "Hard" and "Soft" Label Propagation in Relational Classification

Aram Galstyan
Information Sciences Institute
University of Southern California
Marina del Rey, CA, USA
galstyan@isi.edu

Paul R. Cohen
Information Sciences Institute
University of Southern California
Marina del Rey, CA, USA
cohen@isi.edu

Abstract

Most relational classifiers utilize some sort of label propagation for iterative inference. In this paper we differentiate between "hard" and "soft" label propagation. 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. We 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 [12], iterative label propagation [3, 4], relational neighbor classifiers [6, 7, 8].

While there are various ways to propagate information through relational graphs, in this paper we differentiate between two general classification approaches: In the first, 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 Fig. 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 highest probability of being fraudulent, labels them as fraudulent, and then iterates this procedure.

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, such as class overlap, amount of prior information, and noise in initial class label assignment. 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 noise in initial class label assignment. In other words, *our results suggests that for noisy data, propagating hard*

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

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.

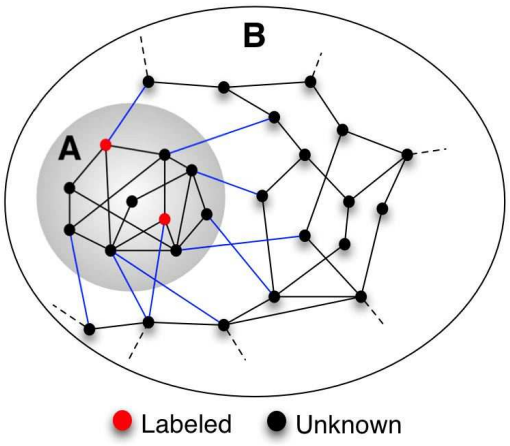

Figure 1. Schematic representation of networked data.

2 Classification Problem

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 attributes (e.g., common author, references, etc.) may be used. As noted earlier, we are mainly interested in relational aspect of classification, so in this paper we ignore intrinsic attributes of data instances and instead examine the effects of relational structures on classification accuracy. We also discard link attributes such as weight, type, and so on. Thus, the data is represented by an undirected binary 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. We want 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. 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 $G(N_b; p_{in}^b)$. 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 $z_{aa}/z_{ab} \equiv z_{in}/z_{out}$ 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}, 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.

Another important issue is the relative sizes of two classes. Indeed, in many real world data *class skew* — in which some classes are much larger than others — can be significant. For instance, in transaction records the number of fraudulent transaction can be orders of magnitude smaller than the total number of transactions. In all but one of the experiments on synthetic data reported in this paper we used skewed class distributions, choosing $N_b = 10N_a$. Because in our model of synthetic data each link across classes is present with the same probability, the cross-class connectivities of nodes in two classes will be substantially different for large skew factors, i.e., $z_{ba} = N_a/N_b z_{ab}$. To keep the total connectivities of both classes balanced, we allow the nodes in class B to have more within-class links, by choos-

ing $p_{in}^b = p_{in}^a + p_{out}(1 - N_a/N_b)$.

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 [12, 13]. 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:

$$s_i^{t+1} = s_{0,i} + \alpha_i s_i^t + \beta \sum_j W_{ij} s_j^t \qquad (1)$$

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, $s_{0,i}$ 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], $s_{0,i} = 0$, $\alpha_i = \alpha$ 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 $s_{0,i} = \alpha_i = 0$, and $\beta_i = 1/z_i$. The resulting updating scheme is:

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

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. [14].

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 to their scores is not very sensitive to the choice of this threshold. After the iteration, the nodes are ranked according to their scores, so that a higher score means a lower rank. This final ranking is then used to calculated the accuracy, as described below.

We need to make the following remark: because of the factor $1/z_i$ in Equation 2, we observed the SP scheme can be sensitive to inbalances in average node connectivities between the two groups. Indeed, if the average connectivity of B nodes is small compared to A nodes, then B-nodes will tend to have higher scores, which will deteriorate classification accuracy. This problem is not present when a uniform normalizing factor independent of the node's degree is chosen, although in this case an opposite bias might affect the performance too [2] . Our experiments, however, suggest that for graphs with balanced connectivity considered here, the model as given by Equation 2 achieves slightly better classification accuracy.

It is important to note that for the model in Equation 2, the scores may be ranked but should not be interpreted as class membership probabilities. For instance, scores $s_i = 0.9$ and $s_j = 0.8$ only suggest that node i is more likely to be in class A then node j. One should not interpret these scores to mean that the node i is in class A with probability 0.9. The SP method ranks nodes rather than classifying them. For classification one needs some kind of thresholding on the scores.

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, $\omega_i^t = \sum_j W_{ij}\sigma^t/z_i$, and then label the nodes for which the fraction is the highest. This procedure is then repeated for T_{max} steps. The pseudo-code of the iterative procedure is shown in Fig. 2

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) trans-

[2] This is because the number of terms that contribute to a node's score are larger for B-nodes.

input weights W_{ij}
input set of initially labeled nodes $\mathcal{A}^0$
input set of initially unlabeled nodes $\mathcal{U}^0$
initialize $\sigma_i \leftarrow 1$, for $i \in \mathcal{A}^0$, $\sigma_i \leftarrow 0$ for $i \in \mathcal{U}^0$
iterate $t = 0 : T_{max}$
    $\omega_{max} \leftarrow 0$
    for all nodes $i \in \mathcal{U}^t$
        $\omega_i^t \leftarrow \sum W_{ij} s_j(t)/z_i$
        if $\omega_{max} < \omega_i^t$ set $\omega_{max} \leftarrow \omega_i^t$
    end for loop
    $\omega_{max}^t = max_i\{\omega_i^t\}$
    for all nodes $i \in \mathcal{U}^t$
        if $\omega_i^t = \omega_{max}$
            $\sigma_i \leftarrow 1$
            $\mathcal{U}^{t+1} \leftarrow \mathcal{U}^t \setminus i$, $\mathcal{A}^{t+1} \leftarrow \mathcal{A}^t \bigcup i$
        endif
    end for loop
end iterate
output $\mathcal{A}^{T_{max}}$
end

Figure 2. Pseudo–code of the label propagation scheme

formation applied after each iteration. This nonlinear transformation constitutes a simple inference process where the class-membership scores of subsets 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, 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, 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 [9]. One of the subtle differences is that [9] use annealing 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 to augment the network with new associations. 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 [11]. The SP model is strongly related to relevance propagation schemes from web-based information retrieval [12, 13]. 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 [5]).

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 [14, 15, 16]. 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, [16] 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 [16] 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. AUC gives the probability that a randomly chosen A node has a lower rank (i.e., higher score) than a randomly chosen B node. An AUC score of 1 corresponds to perfect classification. Let r_i be the rank of the i–th node[3]. Then the AUC score is calculated as follows:

$$AUC = \frac{1}{N_b(N_a - N_a^0)} \sum_{i \in \mathcal{A} \setminus \mathcal{A}^0} \sum_{j \in B} [1 - \theta(r_j - r_i)] \quad (3)$$

where $\theta(x)$ is the step function [4], and the summation over i does not include the initially labeled set.

While the AUC score is a convenient measure for assessing overall quality of ranking, more detailed information can be obtained through ROC analysis. ROC curves describe the relationship between True Positive rates and False Positive rates. They are useful for examining the utility of classification decisions for both symmetric and asymmetric misclassification costs.

For the SP scheme, the ROC curve is obtained by scanning through values for the classification threshold and generating (FP, TP) pairs for each threshold value. For the LP scheme, these pairs can be obtained by scanning through the iteration stopping time from 1 to T_{max}. A more convenient method (which produces identical results) is as follows: Let $n_a(r)$ and $n_b(r)$ be the number of A and B nodes that that have a rank less or equal to r. The ROC curve is then obtained by scanning r from 0 to r_{max}, and generating pairs $(FP(r), TP(r))$ where $FP = n_b(r)/N_b$ and $TP = n_a(r)/(N_a - N_a^0)$.

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 3 we plot the AUC score vs. z_{out}/z_{in} for three different values of z_{in}. The top panel shows the results for the 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.

A similar picture holds in the presence of class skew (bottom panel in Fig. 3). 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. 3). 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. 4.

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

[3]Note that different data instances might share the same rank.

[4]The step function is defined as $\theta(x) = 1, x \geq 0$ and $\theta(x) = 0$ otherwise.

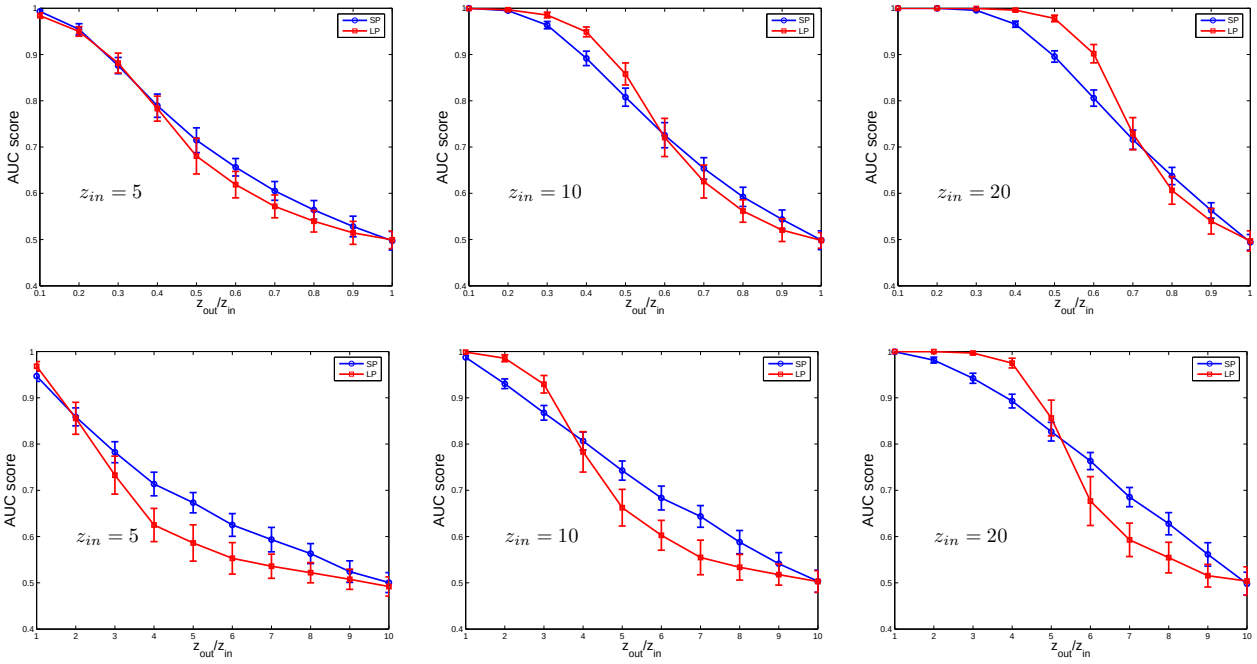

Figure 3. AUC score vs the ratio z_{in}/z_{out} for different values of z_{in}. The top and bottom panels are for equal and skewed class distributions respectively.

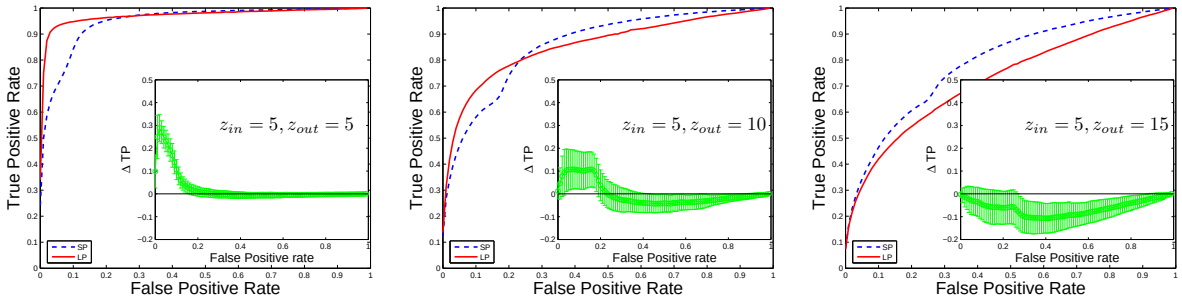

Figure 4. ROC curve for different connectivities.

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). We also note a little "bump" in the SP algorithm's ROC curve: namely, the rate of change for true positives, dTP/dFP, slows down over the FP interval $[0., 0.2]$, and then increases again. This plateau-like structure is even more pronounced for larger values of z_{in}. In fact, for some datasets we observed multiple plateaus, i.e., a "staircases" structure. This indicates some kind of a clustering in the distribution of scores. We indeed observed clusters in the histogram of nodes' scores. We also found that this clustering is due to strong correlation between a node's score and the number of initially labeled nodes it is connected with. Consider two nodes $i \in \mathcal{A}$ and $j \in \mathcal{B}$, and assume they have roughly the same number of neighbors. If j has more links with initially labeled nodes, then after the first iteration it will have a greater score, $s_j^1 > s_i^1$. Our experiments suggest that in majority of cases j will have a higher score even at the end of the iterations.

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.

5.3 Effect of prior knowledge and noise

In the final two sets of experiments, we study the effects on classification accuracy of initially available information and noisy labels. In both experiments we fixed the within-class connectivity to $z_{in} = 10$ and varied the class overlap by choosing different z_{out}.

The top panel in Fig. 5 shows the AUC score for both algorithms as a function of the fraction of the seed nodes N_a^0/N_a for $z_{out} = \{10, 20, 30\}$. Again, each data point was generated by averaging over 100 random realizations. First, we note that for small overlap between the classes both algorithms achieve a remarkable accuracy even with very small fractions of initially labeled nodes. Indeed, for $z_{out} = 10$ and only $\sim 2\%$ of initially labeled nodes, LP achieves near-perfect AUC scores of 0.99, while SP has an AUC score of ≈ 0.95. For larger overlap, the accuracy of both algorithms consistently improve with the amount of available information (i.e., initially labeled seed nodes), although for SP this improvement seems more gradual. We also note that increasing the overlap leads to large fluctuations in the accuracy of the LP algorithm. Indeed, for $z_{out} = 30$ and $N_a^0/N_a = 0.025$, LP has an average AUC score of 0.65 and standard deviation of 0.13, or 20% of the average.

Noise 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. 5 (bottom panel) where we plot the AUC score vs. the ratio N_b^0/N_a^0; again for three different values of class overlap. Clearly, for small 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.

It is worthwhile to examine the different effect of noise on both methods in more details since it is in some way indicative of principal differences between two approaches. Below we provide a simple analytical framework to qualitatively examine this difference. Let us start with SP. Our approach is based on the observation that, as we mentioned earlier, a node's final score is strongly correlated with the number of labeled instances among its immediate neighbors. Indeed, we observed that if between a B-node i and an A–node j the former has more links with the initially

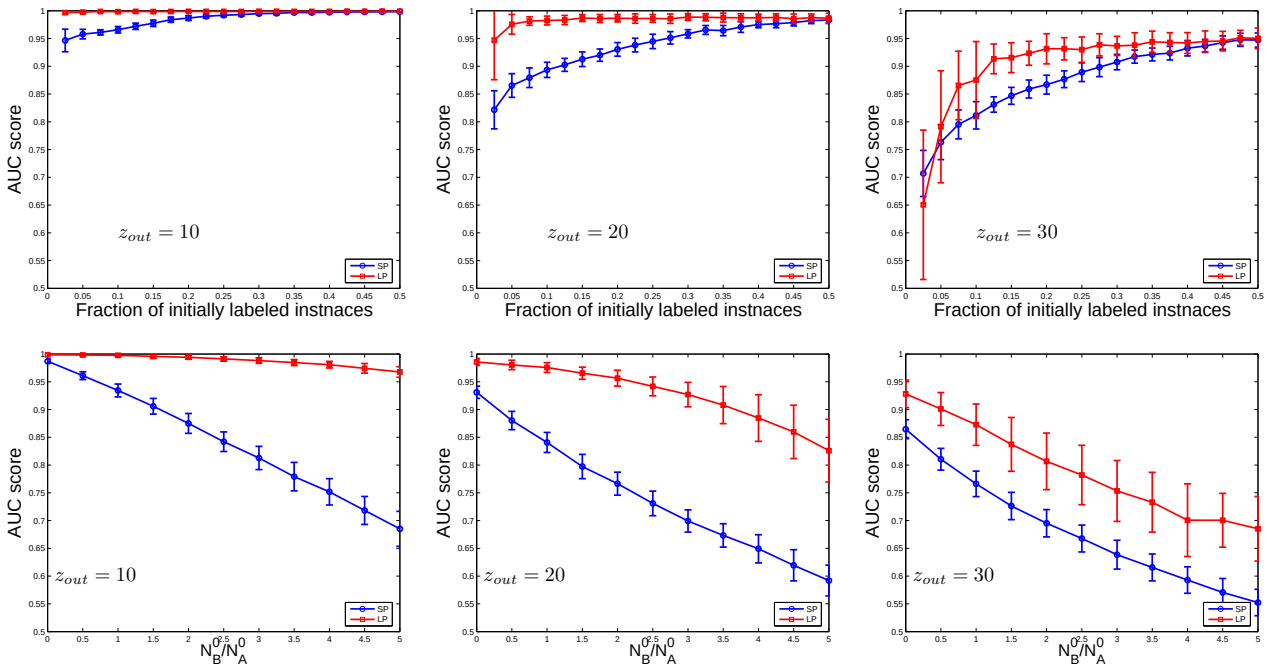

Figure 5. Top panel: AUC score vs the fraction of initially labeled instances (top panel) and number of initially misclassified nodes (bottom panel).

labeled nodes, then in an overwhelming majority of cases i's final score will be greater than j's, contributing negatively to the AUC score. Strictly speaking, due to the factor $1/z_i$ in our model Equation 2 it would be more appropriate to consider the weight rather the absolute number of links with the labeled nodes. For the sake of simplicity, however, we can approximate z_i by its average over the nodes.

Let $P_a(k)$ and $P_b(k)$ be the probability distributions that a randomly chosen node of type A or B is connected to exactly k initially labeled nodes. Then the probability that a randomly chosen A–node has at least as many initially labeled neighbors as a B–node is

$$p_{SP} = \sum_{k=0}^{\infty} P_a(k) \sum_{j=0}^{k} P_b(j) \tag{4}$$

Note that if the assumption above (e.g., higher k means higher score) was always true, then p_{SP} would give us an upper bound on the AUC score.

Let us now consider the LP scheme. It is easy to see that the accuracy of the LP algorithm is determined mostly by the *tails* of the distributions rather than their overlap. Indeed, let $\mathcal{K}^a = \{k_a^1, k_a^2, ...k_a^{N_a - N_0}\}$ and $\mathcal{K}^b = \{k_b^1, k_b^2, ...k_a^{N_b}\}$ be random samples from distributions $P_a(k)$ and $P_b(k)$, respectively, and let $K_{max}^{a,b} = max_k\{k \in \mathcal{K}^{a,b}\}$. K_{max}^a and K_{max}^b are random variables themselves and are distributed according to the *largest order statistic*, $\mathcal{P}_{a,b}(K_{max})$. Unlike the SP case above, we cannot directly obtain an approximation for the AUC score using these distributions. However, we can still examine the effect of the noise by calculating the probability that for a given number of labeled A and B nodes at a certain point in the iterations, no B node will be mislabeled at the next iteration. This probability is given by an equation similar to Eq. 4 with $P_{a,b}(k)$ replaced by respective largest order statistic's distribution,

$$p_{LP} = \sum_{k=1}^{\infty} \mathcal{P}_a(k) \sum_{j=0}^{k-1} \mathcal{P}_b(j) \tag{5}$$

For illustration, we approximate $P_a(k)$ and $P_b(k)$ by Poisson distributions with means λ_a and λ_b, respectively. Assuming N_a^0 and N_b^0 initially labeled nodes for each class, these means can be estimated as $\lambda_a = z_{aa}N_a^0/N_a + z_{ab}N_b^0/N_b$, and $\lambda_b = z_{bb}N_b^0/N_b + z_{ba}N_a^0/N_a$. In Figure 6 we plot p_{SP} and p_{LP} vs the number of mislabeled B–nodes, for $N_a^0 = 40$, $z_{aa} = z_{ab} = 10$, $z_{bb} = 19$, and $z_{ba} = 1$. One can see that p_{SP} decreases with noise almost linearly starting from even very moderate values of N_b^0. This is certainly consistent with the behavior of the AUC score from Figure 5. The behavior of p_{LP}, on the other hand, is rather different: it stays very close to 1 for $N_b^0 < 20$, and then starts to decrease super–linearly, attaining a value ~ 0.75 at the noise level $N_b^0 = 100$. We want to reiterate that p_{LP}

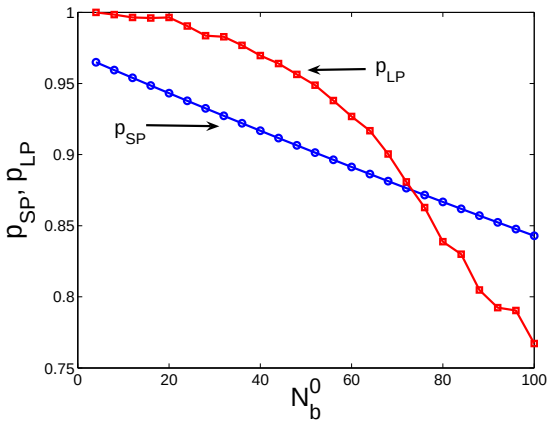

Figure 6. p_{SP} and p_{LP} vs number of mislabeled B nodes.

and p_{SP} should not be compared directly as they stand for different things. While p_{SP} is an approximate upper bound for the AUC score, p_{LP} is simply the probability that at a give iteration step no B node will be labeled. Hence, p_{LP} provides only indirect evidence on the classification accuracy. In particular, the value $p_{LP} \approx 0.75$ simply means that there is a $\sim 25\%$ probability that a B node will be labeled at the first step of the iteration. However, there will also be a (possibly large) number of A nodes labeled at the same iteration step, so the effect of one mislabeled node might be very small on the rest of the iteration process. The fact that the LP algorithm is so resilient to noise clearly suggests that this is the case.

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 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 [10]. 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 topic separately. The parameters of relational graphs corresponding to each classification problem are shown in Table 1.

Topic	N_a	z_{in}	z_{out}
Case–Based	396	–	–
Genetic Algorithms	545	–	–
Probabilistic Methods	483	–	–
Neural Networks	984	–	–
Reinforcement Learning	333	–	–
Rule Learning	176	–	–
Theory	468	–	–

Table 1. Parameters of classification problem for Machine Learning topics. Last two columns show the average connectivities within and across the classes for corresponding topic. Do this over again

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 the majority 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 7 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 8 we show the ROC curves for the same topics. Again, four out of six topics demonstrate behavior that is qualitatively very similar to that presented in Figure 4 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").

7 Discussion and Future Work

Whether one propagates hard labels or scores and probabilities, the main assumption behind these iterative approaches is that true class labels or scores propagate faster than false ones. Our experiments suggest that for relational data with sufficiently strong homophily this is indeed the case, and both algorithms were able to achieve good accuracy. We also found that while neither of approaches dominate over entire range of input parameters, there are some important differences that should be taken account to decide which scheme 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 distinctly 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 positives. This suggests SP might be a better choice for classification purposes only when the cost of missed detections strongly outweigh the cost of false alarms. Not also that this difference will 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 other important finding of this paper is the different behavior of two schemes in the presence noise. Our experiments for synthetic data, as well as for the majority of CoRA topics, suggest that LP algorithm is less sensitive to mislabeled data instances, so that propagating hard labels instead of scores might be a better choice when the prior information is noisy. Although we explained this phenomenon qualitatively, we intend to do more analytical studies to get better understanding.

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 studied this role for fairly simple label and score propagating schemes and a simple graph topology. We believe it would be worthwhile to perform similar studies for more sophisticated classification schemes. We would also like to extend the empirical framework presented here to more complex relational domains that allow both nodes and links to have intrinsic attributes. Currently, evaluations of various relational learning algorithms are limited to a handful of real world datasets. While it is important for an algorithm to perform well on real world data, we believe that conducting controlled set of experiments on synthetic data will help to understand the strengths and weaknesses of the method.

References

[1] S. Chakrabarti, B. E. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In L. M. Haas and A. Tiwary, editors, *Proceedings of SIGMOD-98, ACM International Conference on Management of Data*, pages 307–318, Seattle, US, 1998. ACM Press, New York, US.

[2] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Proceedings of the IJCAI-1999*, pages 1300–1309, 1999.

[3] A. Galstyan and P. R. Cohen. Inferring useful heuristics from the dynamics of iterative relational classifiers. In *Proceedings of IJCAI-05, 19th International Joint Conference on Artificial Intelligence*, 2005.

[4] A. Galstyan and P. R. Cohen. Relational classification through three–state epidemic dynamics. In *Proceedings of the 9th International Conference on Information Fusion*, Florence, Italy, 2006.

[5] Z. Gyongyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *In Proceedings of the 30th VLDB Conference*, 2004.

[6] S. Macskassy and F. Provost. 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] S. Macskassy and F. Provost. 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] S. Macskassy and F. Provost. Netkit-srl: A toolkit for network learning and inference. In *Proceeding of the NAACSOS Conference*, 2005.

[9] S. Macskassy and F. Provost. 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] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval Journal*, 3:127–163, 2000.

[11] 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.

[12] T. Qin, T.-Y. Liu, X.-D. Zhang, Z. Chen, and W.-Y. Ma. 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*, pages 408–415, New York, NY, USA, 2005. ACM Press.

[13] A. Shakery and C. Zhai. Relevance propagation for topic distillation uiuc trec 2003 web track experiments. In *TREC*, pages 673–677, 2003.

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

[15] N. Tishby and N. Slonim. Data clustering by markovian relaxation and the information bottleneck method. In *NIPS*, pages 640–646, 2000.

[16] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Carnegie Mellon University, 2002.

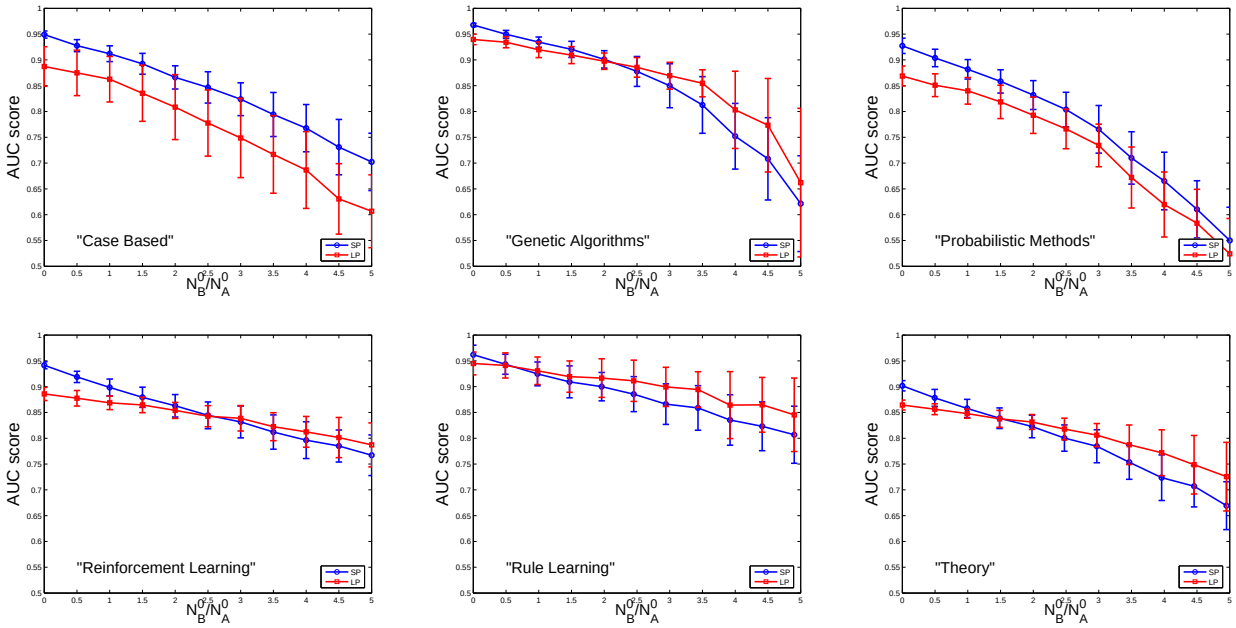

Figure 7. AUC score vs the fraction of initially misclassified nodes for CoRA topics.

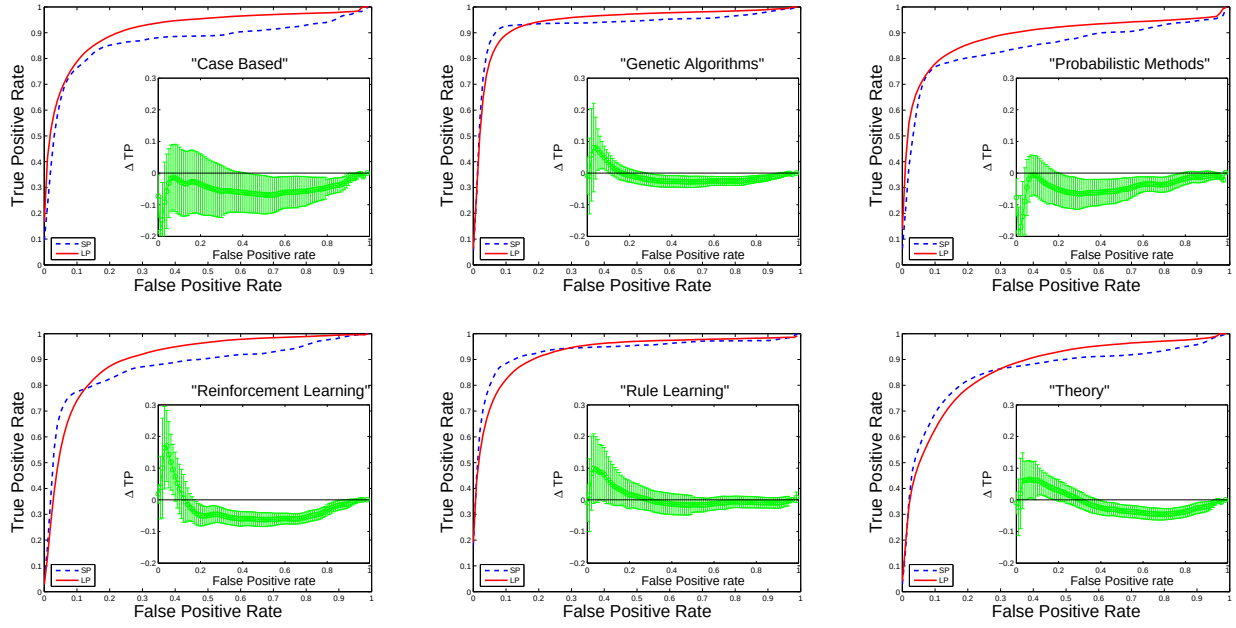

Figure 8. ROC curve for different CoRA topics.