Comparing Diffusion Models for Graph–Based Semi–Supervised Learning

Aram Galstyan GALSTYAN@ISI.EDU
Paul R. Cohen COHEN@ISI.EDU
USC Information Sciences Institute, Marina del Rey, CA, USA

Keywords: graph–based semi–supervised learning

1. Introduction

The main idea behind graph-based semi–supervised learning is to use pair–wise similarities between data instances to enhance classification accuracy (see (Zhu, 2005) for a survey of existing approaches). Many graph–based techniques use certain type of regularization that often involve a graph Laplacian operator (e.g., see (Belkin et al., 2006)). Intuitively, this corresponds to a diffusion process on graphs, where the information is propagated from the labeled instances to the rest of the nodes. Usually, this information is represented as a continuos class–membership probabilities (or scores), and the propagation process corresponds to the diffusion of those scores through the graph.

We contrast this type of continuos diffusion approach by a closely related, but a different one. Specifically, instead of heat–diffusion like process, we consider a discrete, epidemic–like process, where one propagates categorical variables (such as class labels) rather than class membership probabilities. This can be done by combining the diffusion operator with a non–linear transformation that maps the class probabilities onto class labels at each iteration step. We refer to the diffusion and epidemic based approaches as Score Propagation (SP) and Label Propagation (LP), respectively.

Here we compare the two approaches on a semi–supervised learning (ranking) task. Our main findings can be summarized as follows. We find that, while neither approach dominates the other on the entire range of the parameters, there are some interesting differences and tradeoffs between them. Specifically, our results suggest that the epidemic propagation mechanism (LP) tends to be more robust to noise. We also find that, even when the ranking accuracy of both mechanisms are similar in terms of their AUC (area under the curve) scores, they might have significantly different ROC (Receiver–Operator Characteristics) curves, especially for small false positive rates. Below we provide a more detailed account of our findings.

2. The Problem and the Algorithms

We assume that the data is represented as an undirected (symmetric) graph, where the nodes belong to one of two classes, A and B. Given this graph, and a set of initially labeled nodes (queries, or *seeds*) from class A, the task is to rank thee remaining nodes according to their similarity to A.

Continuous Diffusion (SP): The continuous diffusion mechanism used here is very similar to models widely studied in the literature (Zhu & Ghahramani, 2002; Zhou et al., 2004). Let us associate a score s_i with node i, that describes its relative likelihood of being in class A. Then the scores are updated iteratively as follows: $s_i^{t+1} = \frac{1}{z_i}\sum_j W_{ij} s_j^t$. where z_i is the number of neighbors of node i, and W is the adjacency matrix: $W_{ij} = 1$ if nodes i and j are connected and $W_{ij} = 0$ otherwise. Thus, at each iteration, the class membership score of a node is set to the average of the class–membership scores of its neighbors at the previous iteration. The scores of the initially labeled A nodes are *clamped* to 1, while the rest of the nodes are initially assigned a score 0. Because of the clamping, the average score in the system increases with time. Indeed, using the analogy with a heat–diffusion system, it is easy to see that the seed nodes act as diffusion sources that pump more *heat* into the network at each iteration. And since there are no sink–nodes to *absorb* the generated heat, the score of all nodes will eventually converge to 1, provided that the graph is fully connected. To prevent this from happening, we stop the iteration after the average score exceeds some predefined 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.

Epidemic Model (LP): For the epidemic process (LP), we use a simple mechanism that is in some sense the discrete 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 iteration, and for each unlabeled node, we calculate the fraction of the labeled nodes among its neighbors, $\omega_i^t = \sum_j W_{ij}\sigma_j^t/z_i$, find the nodes for which the fraction is the highest, and label them as type A. This procedure is then repeated until all the nodes has been labeled.

While ranking nodes in the SP scheme is straightforward, we need a different ranking mechanism for the LP scheme. Specifically, we assume that the nodes that are similar to the initially labeled nodes will tend to be better connected with them, hence they will be *infected* earlier in the iteration. Thus, we will rank nodes according to their infection times.

3. Main Findings

We have compared two algorithms for a wide range of the parameters, varying class overlap, skew in class sizes, noise in the initially labeled set, and so on. For a full account of those experiments, including ones on real–world data, we refer the reader to (Galstyan & Cohen, 2007). Here we describe two of the most illustrative experiments conducted on synthetic data. The data is generated as follows: The link structure within both classes are described by the Erdos–Renyi graphs $G(N_A; p_{in}^a)$ and $G(N_B; p_{in}^b)$[1], where N_A and N_B are the number of nodes in respective classes. The overlap across the classes is provided by linking each of the $N_A N_B$ possible $A - B$ pairs with probability p_{out}. Thus, 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 $z_{ba} = p_{out} N_A$. Note that if the sizes of two classes are not equal then $z_{ab} \neq z_{ba}$.

ROC analysis In Figure 1(a) we present the ROC analysis of both schemes. For this particular choice of the connectivities, the AUC scores are 0.95 ± 0.01 and 0.97 ± 0.01 for SP and LP, respectively. What is remarkable, however, is that despite the similar overall ranking accuracy, the two classifiers are quite distinct for small false positive rates. In other words, the *difference in the AUC scores is not distributed equally over the whole ROC plane*. Instead, the main difference is for the false positive range $0 < FP < 0.1$. For

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

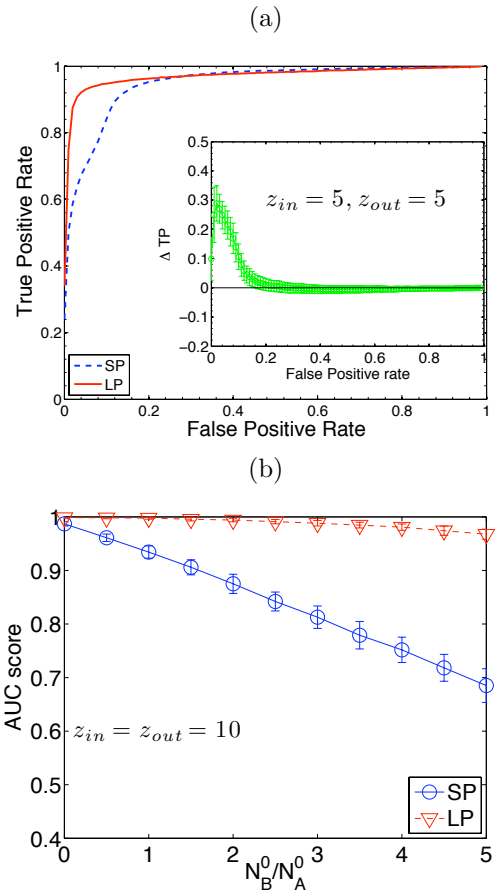

Figure 1. (a) The ROC curves for the LP and SP mechanisms. (b) The AUC score plotted against the ratio N_B^0/N_A^0.

$FP > 0.3$, 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 a highly skewed class distribution, where even tiny false positive rates will translate into a large number of falsely classified instances. The inset shows the difference between true positive rates, $\Delta TP = TP_{LP} - TP_{SP}$, as a function of FP. The bars in the plot 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.

Impact of Noise Next, we examine the effect of noise on the performance of both algorithms. Noise was introduced by randomly choosing N_B^0 nodes from the class B and mislabeling them as type A initially. We set the number of initially labeled A nodes to $N_A^0 = 40$, and studied how the AUC score changed

as we increased the number of mislabeled nodes, N_B^0 (the number of nodes in each class are $N_A = 200$ and $N_B = 2000$). The results are presented in Figure 1 (b), where we plot the AUC score against the ratio N_B^0/N_A^0. Remarkably, the noise has distinctly different effects 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 value of class overlap: The AUC score of the SP algorithm decreases almost linearly, while for the LP algorithm the decrease is initially much slower. Finally, for the case when the class–overlap is very large, the noise seems to affect the performance of both algorithms very similarly.

To find out whether the results presented above hold for more realistic data, we have also experimented with the CoRA data–set of hierarchically categorized computer science research papers (McCallum et al., 2000). Specifically, we focused on the papers in the Machine Learning category which comprises of seven different subtopics[2]. We observed that, generally speaking, the results obtained for the CoRA data were somewhat different from the results for the synthetic data. For instance, we found that the ranking accuracies were lower than one would expect for a random Erdos–Renyi topology with corresponding connectivities. However, our experiments indicate that the main results for the synthetic data also hold for some of the CoRA topics. In particular, we established that for the majority of the topics the LP algorithm was indeed more robust to noise. Furthermore, the different ROC behavior of two algorithms present in synthetic data was observed in the CoRA data as well, with LP achieving better accuracy for smaller false positive rates (Galstyan & Cohen, 2007).

4. Discussion

We have presented empirical comparison of two diffusion schemes for graph–base semi–supervised ranking problem. Our results indicate that even when both approaches have the same overall ranking accuracy, their ROC behavior can be drastically different. Specifically, we found that the discrete label propagation (LP) might be a significantly better choice for small values of acceptable false positive rates. Our second important finding is that when the classes are well–separated, the LP scheme is much more robust to the presence of noise in the initially labeled data. Thus, propagating hard labels instead of scores might be a better choice if the prior information is noisy. This is a very general result, so we believe that it might have important implications in many ranking and classification tasks where the labeled examples might be noisy.

As a future work, we would like to understand the different behavior of two algorithms through analyzing statistical properties of the corresponding models. To give an example of such an analysis, let us consider the impact of noise on both models. First, let us consider the SP mechanism. Our initial investigation suggests that for the SP scheme, a node's final score is strongly correlated with the number of initially labeled nodes (seeds) among its immediate neighbors. In particular, if between a B-node i and an A–node j the former has more links with the seed nodes, then in an overwhelming majority of cases i's final score will be greater than j's, thus contributing negatively to the AUC score. To be more specific, let $P_a(k)$ and $P_b(k)$ be the probabilities that a randomly chosen node of type A or B is connected to exactly k seed nodes. Then the probability that a randomly chosen A–node has at least as many neighboring seed nodes as a B–node is $p_{SP} = \sum_{k=0}^{\infty} P_a(k) \sum_{j=0}^{k} P_b(j)$. 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. Thus, the accuracy of the SP algorithm is affected by the amount of overlap of those two distributions, $P_a(k)$ and $P_b(k)$. Indeed, our initial results suggests that, in the presence of noise, the behaviors of p_{SP} and the AUC score are qualitatively very similar.

For the LP scheme, on the other hand, the accuracy is determined mostly by the *tails* of the distributions $P_a(k)$ and $P_b(k)$ 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, one 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 the expression for p_{SP} with $P_{a,b}(k)$ replaced by respective largest order statistics distribution. Again, our initial results sug-

[2]The multi–class problem was mapped onto a binary classification problem for each individual topic.

gest that even this simple analysis explains, at least qualitatively, some of the observed behavior in the presence of noise. We intend to develop more refined analytical approaches for examining both models more thoroughly.

5. Acknowledgments

This research was supported by the U.S. ARO MURI grant W911NF–06–1–0094 at the University of Southern California.

References

Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.*, *7*, 2399–2434.

Galstyan, A., & Cohen, P. (2007). Empirical comparison of *hard* and *soft* label propagation for relational classification. *Proc. of ILP–07*. Corvallis, Oregon.

McCallum, A., Nigam, K., Rennie, J., & Seymore, K. (2000). Automating the construction of internet portals with machine learning. *Information Retrieval Journal*, *3*, 127–163.

Zhou, D., Weston, J., Gretton, A., Bousquet, O., & Schlkopf, B. (2004). Ranking on data manifolds. *Advances in Neural Information Processing Systems*, *16*, 169–176.

Zhu, X. (2005). *Semi-supervised learning literature survey* (Technical Report). Computer Sciences, University of Wisconsin-Madison.

Zhu, X., & Ghahramani, Z. (2002). *Learning from labeled and unlabeled data with label propagation* (Technical Report). Carnegie Mellon University.