Active Sampling of Networks

Joseph J. Pfeiffer III JPFEIFFER@PURDUE.EDU
Jennifer Neville NEVILLE@CS.PURDUE.EDU
Department of Computer Science, Purdue University, West Lafayette, IN

Paul N. Bennett PAUL.N.BENNETT@MICROSOFT.COM
Microsoft Research, Redmond, WA

Abstract

In network classification, a typical assumption is knowledge of all edges when computing the joint distribution of the instances in the network. That is, for an instance in the network, the neighbors of the instance and their attributes are known. Such settings include social networks such as Facebook where a person's friends are known, allowing for prediction of an attribute of the person given the description of their friends. However, in other domains, relationship information may not be available for all nodes in the network due to privacy or legal restrictions or because a cost is associated with determining the connections of a node. For example, it is unreasonable to expect to be able to access the phone records of the entire population when attempting to identify a handful of individuals involved in illegal or fraudulent activities.

We refer to this problem domain as *Active Sampling*, a domain where instances' labels and edges are acquired through an iterative process in order to identify a handful of instances in a network. In this work, we develop this problem domain formally and present methods estimating the probability of an instance being positively labeled using only the previously acquired samples. Furthermore, we extend our methods to allow for collective inference and learned priors and demonstrate the robustness of the techniques on two synthetic and two real-world datasets.

1. Introduction

A common assumption when classifying instances in a relational domain is that the relationships between the instances are readily observable. In many domains this is a reasonable approach; for example, friends and followers on social networks such as Facebook or Twitter are usually publicly available. This information is used to classify the instances collectively by estimating the labels of instances given their neighbors (Taskar et al., 2007; Neville & Jensen, 2007).

In other domains, the assumption of publicly available relationship information is not appropriate. Such a situation occurs when identifying students involved in academic dishonesty at a large universiy; given a single student caught cheating, it would be unethical to have access to emails between *all* students at the university when there is no evidence to support most of the students being involved. Similarly, in the case of investigating securities fraud it would be useful to have phone records for guilty parties (Neville et al., 2005); however, collecting all phone records for all traders could take a considerable amount of time in addition to violating brokers privacy. Furthermore, in some networks determining the relational linkage may come only at a cost. For example, in the AddHealth dataset (Harris & Udry, 2009), students from various schools were interviewed to find their friendships, as well as attributes such as smoking, drinking, and truancy. However, extensive interviewing is costly, and when applied to a real-world scenario such as identifying students in a school who are likely to be smoking in order to intervene, we may wish to minimize the cost while identifying all students in need of intervention.

While the majority of links in these domains are unavailable, there may remain a number of linkages that *can* be used for classification. Specifically, it is expected that during the investigation of a cheating student or a possibly offending broker the linkages for that

Appearing in *Proceedings of the Workshop on Mining and Learning with Graphs (MLG-2012)*, Edinburgh, Scotland, UK, 2012. Copyright 2012 by the author(s)/owner(s).

instance become available. Thus, after an instance has been investigated, not only have we gained its label, we have also gained its neighbors. Likewise, in the AddHealth case, after committing to interview a candidate, we can determine both if the student was a heavy smoker *and* their friendships.

We refer to this class of problems as *Active Sampling*, where the goal is to iteratively draw samples from the network that have high probability of having a specific label. It is distinct from active labeling in that both the value of that label as well as the structure information of linkage are acquired. Once we have acquired an instance's label we can use the new information to update our classifier and utilize the new linkages to help improve our overall view of the network, either adding a node with no known linkages into the known graph or by giving more links to known (but unlabeled) instances in the network. These additional linkages can help reduce our uncertainty about the instances when estimating their labels.

When estimating the labels of instances in a network, a useful approach is the use of *collective inference* to infer the joint probability distribution over the labelings. This stems from the notion of *correlation*, a statistical dependency between the instances that is found in nearly all networks (Neville & Jensen, 2007; Bilgic & Getoor, 2008; Taskar et al., 2007; Gallagher et al., 2008). A key assumption of most collective inference methods is that of *Markov Independence*, i.e., the instances are conditionally independent of the rest of the network given its neighbors. Such an assumption fails to help the problem of active sampling as only linkages known at a given iteration are between previously labeled instances and their neighbors, meaning linkages between two unlabeled instances are not known. To allow for collective inference when estimating the labels of the unlabeled instances, we follow the work of (Gallagher et al., 2008) and utilize the *2-hop* paths in the known network. As unlabeled instances in the network *can* be 2-hops away from one another, this allows for joint inference of the labelings.

In this work we assume there are no other known attributes of the instances aside from the label we are predicting, meaning the predictors only have relational information available. A natural classifier in this setting is the weighted-vote relational neighbor algorithm (wvRN), which has been shown to perform well in networks with correlation between instances despite its apparent simplicity (Macskassy & Provost, 2007). Furthermore, as we iteratively search for instances in the network that are likely to be positive we can estimate a conjugate prior over the unlabeled instances, which when combined with the wvRN likelihood has a posterior which is easily computed. Utilizing this estimate we can efficiently account for the uncertainty over the unlabeled instances in the network.

The contributions of this work can be summarized as:

- Introduction of a new problem setting (Active Sampling);
- Extension of wvRN to allow for collective classification during Active Sampling;
- Priors for handling uncertainty with unlabeled instances;
- Empirical analysis of models on two synthetic networks as well as two schools in the AddHealth dataset, identifying instances which are characterized as 'Heavy Smokers'.

The rest of the paper is summarized as follows. We discuss related work in Section 2 and formally define the problem and domain in Section 3. Next, we discuss the 2-step collective classification in Section 4 while giving details on the various priors in Section 5. Lastly, we give examples over two networks in Section 6 and give conclusions in Section 7.

2. Related Work

Recently, work in relational learning has explored *active labeling* for both learning and collective inference. In that scenario, research focused on acquiring labels that will improve the accuracy of collective inference by considering properties of the *entire* network structure (Kuwadekar & Neville, 2011; Bilgic & Getoor, 2008; Macskassy, 2009). However, this diverges strongly from the notion of active sampling, where we are examining how to acquire nodes while only having part of the network.

Another related technique is *progressive sampling*, which is centered around determining the optimal sample size to use in order to balance between computational costs and accuracy (Provost et al., 1999; Parthasarathy, 2002; Gu et al., 2001). Progressive sampling first uses a small sample in order to learn an initial model and then uses progressively larger and larger samples as model accuracy improves. In active sampling, we do update the model as samples are acquired; however, we cannot sample the 'ideal' amount from an overall population as we can only utilize the instances and edges already acquired. In addition, it is believed that label and structure acquisition is likely to be considerably more time consuming than updates

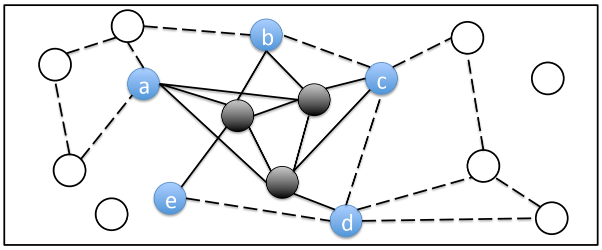


Figure 1. A partially observed network. Black instances known, blue instances have at least one connection to the labeled instances, while white represents separated nodes. Solid lines are known edges, while dashed are unknown.

to the model, as interviewing students concerning their smoking habits and friendships could take several days.

Finally, (Namata et al., 2012) defines on the notion of *active surveying*. In this work, the authors have a given subset of nodes whose labelings are desired, but cannot directly query their labelings. However, the labelings of other nodes in the network can be queried for a cost. The authors then focus on determining which of these nodes should be queried in order to give useful information about the desired subset. This differs from our work as at the start we do not know which nodes we want to have labeled, simply that we wish to discover nodes having a particular label.

3. Problem Formulation

Consider a domain with a graph $G = (V, E)$, where $v \in V$ represents the instances, or people, and $e \in E$ is the edges, or relationships, between the instances. For our model we assume there are no attributes other than y_i for a node v_i, where y_i is the label we are trying to predict. Thus, our estimation must come solely from the correlations with other known instances in the domain. The neighbors of an instance are denoted $\mathcal{N}(v_i)$.

Next, we assume the existence of a sampling mechanism $acquire(v_i)$. We can use *acquire* to obtain information about a node v_i, specifically, *acquire* procures the corresponding label y_i as well as the corresponding relationships $e_{i*} = \{e_{jk} : (j = i \vee k = i) \wedge e_{jk} \in E\}$. The acquisition method is simple to define mathematically, but corresponds with a real-world operation which is likely complex and time-consuming. For the criminal investigation example such a mechanism would need to obtain search warrants to determine the labeling and procure the phone records. This indicates that calling *acquire* more than necessary is undesirable.

The nodes V present in the network can be formally divided into three disjoint sets: Labeled ($\mathcal{I}_L$), Border ($\mathcal{I}_B$) and Separate($\mathcal{I}_S$). The Labeled instances are somewhat self-explanatory - if $v_i \in \mathcal{I}_L$ the label y_i and edges e_{i*} are known. These instances are either given to us at the onset of the problem or obtained iteratively through the *acquire* mechanism. In Figure 1, the darkened nodes in the center are those which belong to the labeled set; note that all of the edges from any node in the labeled set are known regardless of whether the neighbor is in the Labeled set.

The Border instances are those which have at least one known linkage to a labeled instance, that is, if $v_i \in \mathcal{I}_B$ then $\exists\ v_j \in \mathcal{I}_L : e_{ij} \in e_{j*}$. These instances are known to have some sort of relationship with nodes which were previously investigated; for example, if a labeled node was found to be participating in illegal activities it raises the likelihood that its neighbors were as well. The known neighbors of a node v_i are denoted:

$$\mathcal{N}_K(v_i) = \{v_j : (v_i \in \mathcal{I}_L \vee v_j \in \mathcal{I}_L) \wedge (e_{ij} \in E \vee e_{ji} \in E)\}$$

For clarity later, we define $E_K = \forall_{i,j}\{e_{ij} : (v_i \in \mathcal{I}_L \vee v_j \in \mathcal{I}_L) \wedge e_{ij} \in E\}$, or the set of edges which have at least 1 node in the labeled set (and are therefore known). Observe that Border instances cannot be directly connected to one another. In Figure 1 the Border nodes are those which are labeled with letters a-e. The dashed lines indicate edges which are present in the true graph structure, but are unavailable for use. The edges between the Border nodes and the Labeled nodes are known, due to the edges of the Labeled instances having been returned by the *acquire* method.

In contrast, the set of Separate nodes $\mathcal{I}_S$ indicates those which have no known connection to the labeled ones, meaning if $v_i \in \mathcal{I}_S$ then $\mathcal{N}_K(v_i) = \emptyset$. These nodes are denoted with the open circles in Figure 1. Note that the only types of edges these can have are unobserved edges, possibly to the Border nodes or possibly to other Separate nodes.

General Active Sampling Procedure The algorithm used by a general active sampling algorithm can be seen in Algorithm 1. ActiveSample chooses one sample to acquire; recursively calling itself until the total number of samples desired have been obtained: line 25 is the recursive call, while lines 2-4 handle the base case. Once a set of probabilities for the unlabeled instances has been computed (lines 7-8), the maximum of those probabilities is then picked for inclusion in the labeled set (lines 12-17). The *acquire* method is then called to obtain the labeling and the node is inserted in the labeled set (line 20). Lastly, the sets of Border and Separate nodes as well as the set of known edges

Algorithm 1

ActiveSample(G, $\mathcal{I}_L$, $\mathcal{I}_B$, $\mathcal{I}_S$, E_K, ns)

```
 1: # When all samples are acquired
 2: if $|\mathcal{I}_L| = $ ns then
 3:     return $\mathcal{I}_L$
 4: end if
 5:
 6: # Compute Probs for all Unlabeled Instances
 7: bprobs = CompBorderProbs($\mathcal{I}_L$, $\mathcal{I}_B$, $E_K$)
 8: sprob = CompSepProb($\mathcal{I}_L$)
 9:
10: # Choose maximum from Border probabilities;
11: # compare with probability of Separate instances
12: maxbprob = max(bprobs)
13: if maxbprob < sprob then
14:     maxnode = RandomChoice($\mathcal{I}_S$)
15: else
16:     maxnode = CorrespondingNode(maxbprob)
17: end if
18:
19: # Acquire the node
20: $\mathcal{I}_L = \mathcal{I}_L \cup acquire$(maxnode)
21:
22: # Compute new $\mathcal{I}_B$, $\mathcal{I}_S$, $E_K$
23: $[\mathcal{I}_B, \mathcal{I}_S, E_K]$ = ComputeKnownGraph($G$, $\mathcal{I}_L$)
24:
25: return ActiveSample($G$, $\mathcal{I}_L$, $\mathcal{I}_B$, $\mathcal{I}_S$, $E_K$, ns)
```

are updated, and the recursive call is made.

The accuracy of the method thus hinges on lines 7-8: lines which estimate the labelings for both the border instances and separate instances. Next, we discuss several methods for accurately estimating those probabilities, beginning with the weighted vote relational neighbor model for relational classification (Macskassy & Provost, 2007), then extending into more advanced methods such as how to utilize collective inference as well as estimation of priors for each iteration.

4. Weighted Vote and Collective Classification

The key portion of this work is the choice of which node to sample next from the pool of unlabeled instances. Our domain lacks attribute information, meaning we must rely on the known labels of previous samples in order to determine which instance to sample next. A simple and appropriate classifier for this domain is the Weighted Vote Relational Neighbor (wvRN) classifier, defined by (Macskassy & Provost, 2007). The wvRN makes the *Markov* assumption, meaning the variable y_i is conditionally independent of the rest of the graph G, given the neighbors $\mathcal{N}(v_i)$, that is $P\left(y_i|G\right) = P\left(y_i|\mathcal{N}(v_i)\right)$. Building on this, the wvRN defines a simple probability distribution over the neighbors of v_i, defined as:

$$P\left(y_i|\mathcal{N}(v_i)\right) = \frac{1}{Z} \sum_{v_j \in \mathcal{N}(v_i)} w_{ij} \cdot P\left(y_i|y_j\right)$$

As our domain works only within the known edges, our distribution becomes:

$$P\left(y_i|\mathcal{N}_K(v_i)\right) = \frac{1}{Z} \sum_{v_j \in \mathcal{N}_K(v_i)} w_{ij} \cdot P\left(y_i|y_j\right) \qquad (1)$$

We assign the straightforward pdf:

$$P\left(y_i = c|y_j\right) = \begin{cases} 1 & \text{if } y_j = c \\ 0 & \text{otherwise} \end{cases}$$

Assuming binary labels and all weights equal to 1, equation 2 reduces to:

$$P\left(y_i|\mathcal{N}_K(v_i)\right) = \frac{1}{|\mathcal{N}_K(v_i)|} \sum_{v_j \in \mathcal{N}_K(v_i)} y_j \qquad (2)$$

4.1. Collective Classification

As discussed in several works, the usage of collective classification in relational domains can greatly increase the accuracy of estimates for each node (Neville & Jensen, 2007; Taskar et al., 2007). As pointed out earlier, the border instances we are trying to classify are conditionally independent of each other given the labeled instances due to the Markov assumption of the wvRN. However, if we follow the intuition given by (Gallagher et al., 2008) we can used the *2-hop paths* to establish connections between the border instances.

We therefore have the set E^2 consisting of the 2-hop edges, where $e^2_{ij} \in E^2$ is defined as:

$$e^2_{ij} = \bigcup_{k=1}^{|V|} \mathbb{I}\left[(e_{ik} \cap e_{kj}) \in E\right]$$

Where $\mathbb{I}[b]$ returns 1 or 0 depending on if b is true or not true, so e^2_{ij} is simply whether a 2-hop path between v_i and v_j exists. E^2_L is defined similarly, only including the edges that are connected to a labeled instance.The *known* 2-hop neighbors are then denoted $\mathcal{N}^2_K(v_i)$, defined as:

$$\mathcal{N}^2_K(v_i) = \{v_j : e^2_{ij} \in E^2_K\}$$

Using the 2-hop distances, we can rewrite equation 2 as:

$$P\left(y_i|\mathcal{N}^2_K(v_i)\right) = \frac{1}{|\mathcal{N}^2_K(v_i)|} \sum_{v_j \in \mathcal{N}^2_K(v_i)} y_{ij} \qquad (3)$$

While equation 2 has no opportunity for collective classification due to the border nodes being conditionally independent from one another, the 2-hop distribution shown in equation 3 can use collective classification as the Border nodes can be connected to one another. To do this we can use Gibbs sampling to estimate the joint distribution of the Border labels, making the algorithm a variant of Relational Dependency Networks (Neville & Jensen, 2007) with the wvRN defining the conditional distributions.

4.2. Weighted Collective Classification

Lastly, we introduce weighting into the distribution:

$$w_{ij}^2 = \sum_{k=1}^{|V|} \mathbb{I}\left[(e_{ik} \cap e_{kj}) \in E_K\right]$$

Essentially, the weights are now a count of the number of 2-hop paths. For example, in Figure 1 when determining the conditional distribution for node (a), node (b)'s value is given twice as much weight as node (c)'s due to there being two 2-hop paths from (a) to (b) as opposed to a single one from (a) to (c). Our distribution becomes:

$$P_W\left(y_i|\mathcal{N}_K^2(v_i)\right) = \frac{1}{\sum_{v_j \in \mathcal{N}_K^2(v_i)} w_{ij}^2} \sum_{v_j \in \mathcal{N}_K^2(v_i)} w_{ij}^2 y_{ij} \tag{4}$$

This has the effect up upweighting the nodes 2-hops away that share the most common neighbors with v_i. Since nodes with many common neighbors likely share more common interests, it is likely they have a higher probability of sharing the same label.

4.3. Separate Nodes

As we have no attribute or relational information for the Separate nodes, we assign a probability of being positive for the Separate nodes based on the previous draws from the Separate population. Let $\mathcal{S}$ be elements of $\mathcal{I}_L$ which were chosen from the Separate population. Our estimation that the probability a Separate node is positive is:

$$\hat{\theta}_S = \frac{\sum_{v_i \in \mathcal{S}} y_i}{|\mathcal{S}|}$$

4.4. Node Selection

The active sampling (AS) method we use is a combination of the estimates for the border and the separate nodes. The active sampling maximum likelihood method (ASML) simply chooses from the nodes the one with the highest likelihood from either estimation.

5. Handling Uncertainty with Priors

In order to pick instances with less uncertainty, we introduce a Beta prior for both the Border nodes and Separate nodes. To do this, we first estimate the proportion of nodes in the labeled set that are positive:

$$\theta_L = \frac{\sum_{v_i \in \mathcal{I}_L} y_i}{|\mathcal{I}_L|}$$

We now have a predefined weight for the prior γ which controls the weight of the prior with respect to the likelihood of the Separate (or Border) nodes. Our posterior distribution for the Separate θ_S is simply another Beta distribution with expected value:

$$\mathbb{E}\left[\theta_S|\theta_L, \gamma, \mathcal{S}\right] = \frac{\theta_L\gamma + \sum_{v_i \in \mathcal{S}} y_i}{\gamma + |\mathcal{S}|}$$

Thus, when choosing from the set of Border and Separate nodes, we assume all Separate nodes are positive with probability $\mathbb{E}\left[\theta_S|\theta_L, \gamma, \mathcal{S}\right]$.

Next, we need a comparable prior for the Border nodes. As a first step, we create a probability of a positive value over a *random draw* from the border population, this probability of positive value over the population of border instances is then used to formulate a prior for each border instance. Let $\mathcal{B}$ be elements of $\mathcal{I}_L$ which were chosen from the Border population. We get an expected value for θ_B in a similar manner as θ_S:

$$\mathbb{E}\left[\theta_B|\theta_L, \gamma, \mathcal{B}\right] = \frac{\theta_L\gamma + \sum_{v_i \in \mathcal{B}} y_i}{\gamma + |\mathcal{B}|}$$

Now we use the expected value of a positive draw from the border nodes to formulate a prior to be used in conjunction with each individual border node. Formulating it as a Beta prior again, the expected value of the posterior distribution for an individual node $v_i \in \mathcal{I}_B$ is:

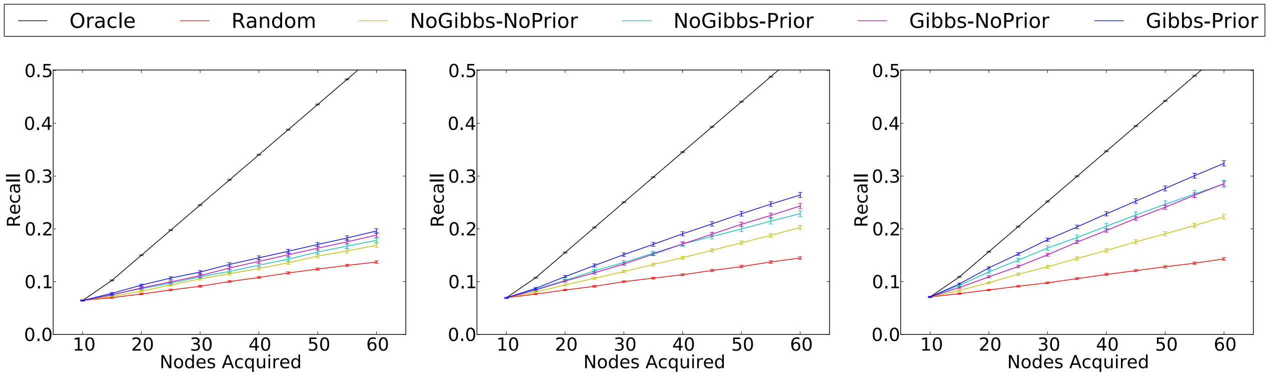
$$\mathbb{E}\left[y_i|E\left[\theta_B\right], \gamma, \mathcal{B}\right] = \frac{E[\theta_B]\gamma + \sum_{v_j \in \mathcal{N}_L^2(v_i)} w_{ij}^2 y_{ij}}{\gamma + \sum_{v_j \in \mathcal{N}_L^2(v_i)} w_{ij}^2}$$

The active sampling maximum expected posterior (ASMEP) then picks the maximum from:

$$\left\{ \bigcup_{v_i \in \mathcal{I}_B} \mathbb{E}\left[y_i|E\left[\theta_B\right], \gamma, \mathcal{B}\right], \bigcup_{v_i \in \mathcal{I}_S} \mathbb{E}\left[\theta_S|\theta_L, \gamma, \mathcal{S}\right] \right\}$$

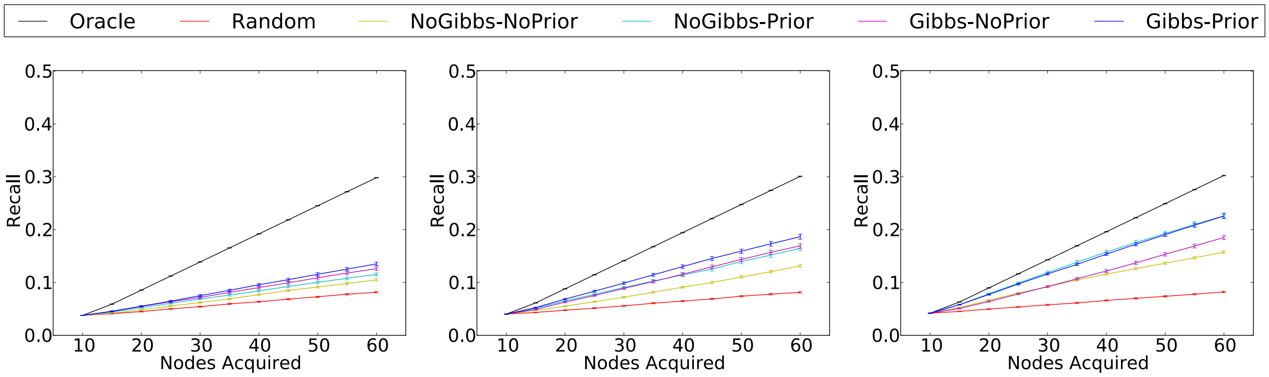
6. Experiments

In this section we begin by generating a large number of labels on top of existing, real-world networks in order to simulate how the methods perform over a distribution of networks. The first network we use is one



(a) AddHealth (.2 AC) (b) AddHealth (.3 AC) (c) AddHealth (.4 AC)

Figure 2. Recall Scores for the first AddHealth school network with synthetic labelings having varying levels of autocorrelation.



(a) Rovira (.2 AC) (b) Rovira (.3 AC) (c) Rovira (.4 AC)

Figure 3. Recall Scores for the RoviraEmail network with synthetic labelings having varying levels of autocorrelation.

of the schools from the AddHealth dataset (Harris & Udry, 2009), while the second we choose for this task is an email network collected from the University of Rovira (Guimera et al., 2003). For each of these real-world networks we generate a set of labelings where approximately one sixth of the labels are positive with the rest being negative, with varying levels of (fairly low) correlation for each (Table 1). For each network we generate a set of 100 different labelings which the methods are then run on, with different starting labels each run.

In addition to the synthetic networks, we chose two of the schools in the AddHealth dataset which have correlations along the Smoking label. The Smoking label has an integer value between 0 and 6; 0 indicating students who never smoke while 6 indicates heavy smokers. We used a threshold for our labeling, denoting 1 for values 4,5, and 6 and 0 denoting the rest. The task is then to examine the heaviest smokers in the school. Similar to the synthetic network we run each method 100 times; however, instead of relabeling the network we simply start the methods from different initial nodes.

Dataset	Nodes	+ Prop	Mean AC	Std AC
AddHealth (School 1)	635	0.24	0.20	N/A
AddHealth (School 2)	576	0.15	0.23	N/A
AddHealth (Synthetic)	635	0.17	0.216	0.03
	635	0.17	0.306	0.056
	635	0.17	0.391	0.077
Rovira (Synthetic)	1,133	0.17	0.204	0.054
	1,133	0.17	0.298	0.054
	1,133	0.17	0.390	0.040

Table 1. Autocorrelation across linkages in the synthetic networks. Broken down by the proportion of nodes which were positive, the average autocorrelation of the network and the standard deviation of the autocorrelations.

6.1. Methods

We compare several methods on each dataset, recording the recall of each as the iterative process regresses. First, we examine two baselines, denoted *Oracle* and *Random*. The *Oracle* method always chooses a pos-

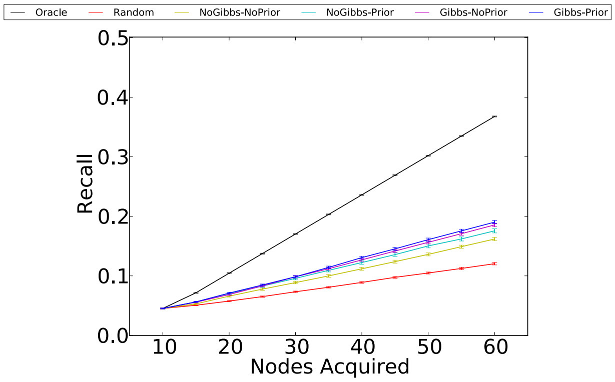


Figure 4. Recall Scores the Smoking attribute the first AddHealth school.

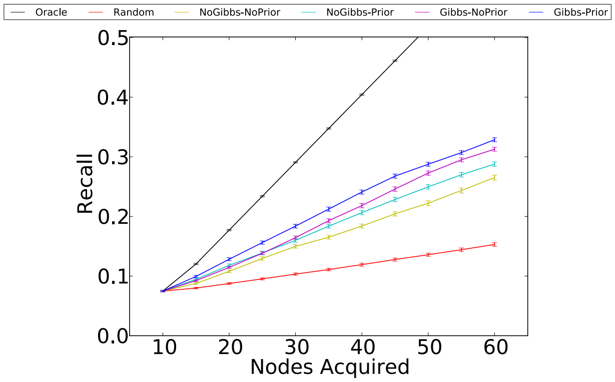


Figure 5. Recall Scores the Smoking attribute on the second AddHealth school.

itive instance from the Border *when available*, otherwise it randomly selects from the Separate set. This represents the best any algorithm can do, should it estimate the positive probability for the border instances perfectly. Additionally, the *Random* baseline simply chooses instances at random from the network. All methods should perform at least as well as *Random*.

For our methods we compare usage of the Gibbs collective classification and the Beta prior with versions that omit either Gibbs, the prior, or both. All algorithms use the weighted version of the 2-hop paths; in addition, we set $\gamma = 2$ for all of the priors. The Gibbs sampler is done over only 100 iterations with the first 50 being considered the "burn-in" and thrown out; this is due to the fact Gibbs must be ran for each iteration, so we trade-off accuracy for efficiency. The initial values are set by a draw from the conditional distributions of the Labeled instances.

For each method we collect 10 'initial' starting points to grow from; 5 are randomly selected positive instances (perhaps students who have been caught smoking) and 1 neighbor of each of the 5 (a friend). We then compare each method on 50 samples, using Recall to determine the method which after 60 total samples has found the highest proportion of positive instances. In the smoking case, the more students we find that are heavy smokers the more the administration can help.

6.2. Analysis

Studying the synthetic networks first (Figures 2,3), we see that the higher amount of correlation in the network results in a higher recall for each of the non-baseline methods (*Oracle* and *Random*). Interestingly, we find that for lower amounts of correlation Gibbs sampling (without priors) outperforms the usage of priors (without Gibbs). However, as correlation increases between the instances the prior becomes more useful than Gibbs. When there is high correlation a few neighbors can likely give a reasonable estimate of the probability for an instance, making the prior more useful for discerning between the amount of confidence in the estimates. In the low correlation case the Gibbs is likely more useful due to its ability to more accurately estimate the joint probability of the instances; not having enough information may result in a very poor estimate, making the prior version less useful (but still better than no prior).

While having either Gibbs sampling or priors improves over omitting both, the combination of the two is the most powerful method. We can see across Figures 2 and 3 that combining the Gibbs sampler with the priors helps in cases where there is little correlation or considerable correlation between the instances. Thus, we can effectively combine them to create a method which has the advantages of both, without detracting from either one.

Next, we run the methods on the AddHealth schools looking for the heavy Smoker labeling (Figures 4,5). Both of these networks have a low correlation (around .2), resulting in the Gibbs sampler (without priors) performing better than the priors (without Gibbs), while both continue to outperform omission of Gibbs and a prior. Furthermore, we again see that combination of the Gibbs sampling and priors outperforms either by itself.

7. Conclusions

In this work we have introduced the problem of Active Sampling, where nodes are investigated and labeled without full access to the entire network. We have shown the promise of utilizing a weighted vote relational neighbor model over the squared network, allowing for collective classification of the Border instances. Furthermore, a method for learning priors over the Border and Separate nodes was demonstrated to have a significant increase in recall when applied to a distribution of networks.

Future work on this area is extensive as considerable effort can be made towards balancing short term gain with long term gain. Additionally, this work covers networks with no attributes aside from the label we are trying to predict; incorporation of additional attributes could improve overall accuracy.

Acknowledgements

This research is supported by NSF under contract numbers IIS-1017898 and IIS-1149789. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of NSF or the U.S. Government.

We thank Timothy La Fond and Sebastian Moreno for help with data collections and useful discussions.

References

Bilgic, Mustafa and Getoor, Lise. Effective label acquisition for collective classification. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pp. 43–51, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4.

Gallagher, Brian, Tong, Hanghang, Eliassi-Rad, Tina, and Faloutsos, Christos. Using ghost edges for classification in sparsely labeled networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pp. 256–264, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4.

Gu, Baohua, Liu, Bing, Hu, Feifang, and Liu, Huan. Efficiently determining the starting sample size for progressive sampling. In De Raedt, Luc and Flach, Peter (eds.), *Machine Learning: ECML 2001*, volume 2167 of *Lecture Notes in Computer Science*, pp. 192–202. Springer Berlin / Heidelberg, 2001. ISBN 978-3-540-42536-6.

Guimera, R., Danon, L., Diaz-Guilera, A., Giralt, F., and Arenas, A. Self-similar community structure in a network of human interactions. *Phys. Rev. E*, 68: 065103, Dec 2003.

Harris, K.M., C.T. Halpern E. Whitsel J. Hussey J. Tabor P. Entzel and Udry, J.R. The national longitudinal study of adolescent health: Research design [www document], 2009.

Kuwadekar, Ankit and Neville, Jennifer. Relational active learning for joint collective classification models. In Getoor, Lise and Scheffer, Tobias (eds.), *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pp. 385–392, New York, NY, USA, June 2011. ACM. ISBN 978-1-4503-0619-5.

Macskassy, Sofus A. Using graph-based metrics with empirical risk minimization to speed up active learning on networked data. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pp. 597–606, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9.

Macskassy, Sofus A. and Provost, Foster. Classification in networked data: A toolkit and a univariate case study. *J. Mach. Learn. Res.*, 8:935–983, December 2007. ISSN 1532-4435.

Namata, Galileo, London, Ben, Getoor, Lise, and Huang, Bert. Query-driven active surveying for collective classication. In *10th International Workshop on Mining and Learning with Graphs*, 2012.

Neville, Jennifer and Jensen, David. Relational dependency networks. *Journal of Machine Learning Research*, 8:2007, 2007.

Neville, Jennifer, Şimşek, Özgür, Jensen, David, Komoroske, John, Palmer, Kelly, and Goldberg, Henry. Using relational knowledge discovery to prevent securities fraud. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pp. 449–458, New York, NY, USA, 2005. ACM. ISBN 1-59593-135-X.

Parthasarathy, Srinivasan. Efficient progressive sampling for association rules, 2002.

Provost, Foster, Jensen, David, and Oates, Tim. Efficient progressive sampling. In *In Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, pp. 23–32. ACM Press, 1999.

Taskar, B., Abbeel, P., Wong, M.-F., and Koller, D. Relational markov networks. In Getoor, L. and Taskar, B. (eds.), *Introduction to Statistical Relational Learning*. MIT Press, 2007.