Effective Label Acquisition for Collective Classification

Mustafa Bilgic
Computer Science Dept.
University of Maryland
College Park, MD 20742
mbilgic@cs.umd.edu

Lise Getoor
Computer Science Dept.
University of Maryland
College Park, MD 20742
getoor@cs.umd.edu

ABSTRACT

Information diffusion, viral marketing, and collective classification all attempt to model and exploit the relationships in a network to make inferences about the labels of nodes. A variety of techniques have been introduced and methods that combine attribute information and neighboring label information have been shown to be effective for collective labeling of the nodes in a network. However, in part because of the correlation between node labels that the techniques exploit, it is easy to find cases in which, once a misclassification is made, incorrect information propagates throughout the network. This problem can be mitigated if the system is allowed to judiciously acquire the labels for a small number of nodes. Unfortunately, under relatively general assumptions, determining the optimal set of labels to acquire is intractable. Here we propose an acquisition method that learns the cases when a given collective classification algorithm makes mistakes, and suggests acquisitions to correct those mistakes. We empirically show on both real and synthetic datasets that this method significantly outperforms a greedy approximate inference approach, a viral marketing approach, and approaches based on network structural measures such as node degree and network clustering. In addition to significantly improving accuracy with just a small amount of labeled data, our method is tractable on large networks.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology—*Classifier design and evaluation*

General Terms

Algorithms

Keywords

active inference, label acquisition, collective classification

KDD'08, August 24–27, 2008, Las Vegas, Nevada, USA.
Copyright 2008 ACM 978-1-60558-193-4/08/08 ...$5.00.

1. INTRODUCTION

Information diffusion, viral marketing, and collective classification all attempt to exploit relationships in a network to reason and make inferences about the nodes in the network. The common intuition is that knowing (or inferring) something about the label of a particular node can tell us something useful about the other nodes in the network. For instance, the labels of the linked nodes tend to be correlated (not necessarily a positive correlation) for many domains; hence, finding the correct label of a node is useful for not only that particular node, but it also has an impact on the predictions that are made about the rest of the network. Thus, it has been shown that methods such as collective classification, i.e., classifying the nodes of a network simultaneously, can significantly outperform the traditional independent labeling approaches [2, 7, 13, 15, 24, 26].

However, sometimes, the advantage of exploiting the relationships can become a disadvantage. An incorrect prediction about a particular node (due to an approximate inference procedure, noise in the data, model limitations, etc.) can propagate in the network and lead to incorrect predictions about other nodes. For example, consider a simple binary classification problem where an island of nodes that should be labeled with the positive label are surrounded with a sea of negatively labeled nodes. The island may be flooded with the labels of the neighboring nodes in the sea; this can happen for example when the model prefers intra-class interactions over inter-class interactions and achieves this objective by flooding the island nodes with the labels of the nodes in the sea.

This problem can be alleviated if the collective inference algorithm is allowed to judiciously acquire labels for a small set of nodes in the network. Depending on the application, labels can be acquired by asking users to rate specific items, a company can provide free samples to a small set of customers and customers' viral networking or purchasing behavior can be observed, or laboratory experiments can be performed to determine protein functions, etc. However, as we show later, determining the optimal set of labels to acquire is intractable under relatively general assumptions. Therefore, we are forced to resort to approximate and heuristic techniques to get practical solutions.

In this paper, we describe three polynomial-time label acquisition strategies. The first and most obvious approach is based on approximating the objective function (which we define formally in Section 2) and greedily acquiring the label that provides the highest improvement in the objective value. The second approach is a direct application of a viral marketing model. The third approach is a simple yet effective acquisition method that learns the cases when a given collective classification model makes mistakes, finds islands of nodes that the collective model is likely to misclassify, and suggests acquisitions to correct these potential mistakes.

We compare these three methods to one another and also to acquisition strategies that are based on network structural measures such as node degree and network clustering. We empirically show that the third method we propose significantly outperforms all of the other methods on both real and synthetic datasets.

The label acquisition problem has received ample attention within the context of active learning [3, 16, 27]. There are two main differences between the scenario we address and the active learning scenario. First, active learning has traditionally been concerned with flat data; here, we are interested in network data. The second and the biggest difference is that we assume that we have available an already trained model of the domain, and thus the learning has been done offline, but we have the option to acquire labels to seed the classification during inference. This is the setting Rattigan et al. [22] introduced and referred to as "active inference." They looked at the relational network classifier, introduced by Macskassy and Provost [14] in which there are no node attributes; only labels are propagated. Here, we build on this, and look at networks in which the nodes have attribute information and compare to the structural strategy that they introduced.

Our contributions in this paper are:

- We empirically show that the most obvious method does not perform well in practice.
- We show a mapping between the viral marketing problem and the label acquisition problem.
- We propose an acquisition method that learns when a given collective classification model makes mistakes and suggests acquisitions to correct possible mistakes.
- We empirically show that this method outperforms all other methods.

We next formulate the label acquisition problem and state the objective function in Section 2. Then, we explain the three approaches in Sections 3, 4, and 5. We then show experimental results on both synthetic and real datasets (Section 6). We finally discuss related work (Section 7) and future work (Section 8) and then conclude (Section 9).

2. PROBLEM FORMULATION

In this section, we review the collective classification problem and define the objective function for label acquisition for collective classification. In this problem, we assume that our data is represented as a graph with nodes and edges, $G = (\mathcal{V}, \mathcal{E})$. Each node $V_i \in \mathcal{V}$ is described by an attribute vector X_i and a class label Y_i pair, $V_i = \langle X_i, Y_i \rangle$. The X_i is a vector of individual attributes $\langle X_{i1}, X_{i2}, \ldots, X_{ip} \rangle$ and the domain of Y_i is $\{y_1, y_2, \ldots, y_m\}$. Each edge $E_{ij} \in \mathcal{E}$ describes some sort of relationship between its endpoints, $E_{ij} = \langle V_i, V_j \rangle$.

Examples include: 1) social networks, where the nodes are people, the attributes include demographic information such as age and income and the edges are friendships: we may be interested in labeling the people that are likely to partake in some behavior (e.g., smoking, IV drug use) or have some disease (e.g., a sexually transmitted disease, obesity), 2) citation networks, where the nodes are publications, the attributes include some content information and the edges are the citations, and we may be interested in finding seminal papers or categorizing the topics of the papers, and 3) biological networks, where the nodes are proteins, attributes include annotations, edges represent interactions, and we may be interested in inferring protein function.

2.1 Collective Classification

Often in graph data, the labels of nodes are correlated (though not necessarily positively correlated). For example, friends tend to have similar smoking behaviors, papers are likely to share the same topic of the papers that they cite, proteins are likely to have complementary functions. *Collective classification* is the term used for simultaneously predicting the labels $\mathcal{Y}$ of $\mathcal{V}$ in the graph G, where $\mathcal{Y}$ denotes the set of labels of all of the nodes, $\mathcal{Y} = \{Y_1, Y_2, \ldots, Y_n\}$.

In general, the label of a node can be influenced by its own attributes and the labels and attributes of other nodes in the graph. There are many collective classification models proposed to date that make different modeling assumptions about these dependencies. For instance, Neville and Jensen [19], Lu and Getoor [13], Macskassy and Provost [15], and McDowell et al. [18] make use of local models, such as Naive Bayes, Logistic Regression, etc., as a function of the local attributes X_i and aggregation of the neighbor labels. Chakrabarti et al. [2] considered using the local attributes of the neighboring nodes and showed that it in fact hurts the overall performance. Taskar et al. [26] fit a Markov random field, where the labels and attributes are the random variables. For the purposes of this paper, we take this latter approach and describe it formally.

Let $\mathcal{N}_i$ denote the labels of the neighboring nodes of V_i, $\mathcal{N}_i = \{Y_j | \langle V_i, V_j \rangle \in \mathcal{E}\}$. We make the common Markovian assumption that Y_i is directly influenced only by X_i and $\mathcal{N}_i$. Given the values of $\mathcal{N}_i$, Y_i is independent of $\mathcal{Y} \setminus \mathcal{N}_i$ and is independent of $\mathcal{X} \setminus \{X_i\}$, where $\mathcal{X}$ denotes the set of all attribute vectors in the graph, $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$. The joint probability of $P(\mathcal{Y}|\mathcal{X})$ is then given by:

$$P(\mathcal{Y}|\mathcal{X}) = \frac{1}{Z} \prod_{Y_i \in \mathcal{Y}} \phi(Y_i, \mathcal{N}_i) \phi(Y_i, X_i)$$

where ϕ are compatibility functions, and Z is the partition function. Note that these compatibility functions, as we defined them, can be 3 or more dimensional. One simplifying assumption is to assume a pairwise MRF [26]. Then,

$$P(\mathcal{Y}|\mathcal{X}) = \frac{1}{Z} \prod_{Y_i \in \mathcal{Y}} \left(\prod_{j=1}^{p} \phi(Y_i, X_{ij}) \right) \left(\prod_{Y_j \in \mathcal{N}_i} \phi(Y_i, Y_j) \right)$$

We do not discuss the details of defining the potential functions. The interested reader can find more details in [26].

We also assume that we are given a training graph $G^{tr}(\mathcal{V}^{tr}, \mathcal{E}^{tr})$ where all of the labels are known. We train our collective model, CM, using this training graph. Given the test graph G, a trained model CM, and assuming the values of the attribute vectors $\mathcal{X}$ are known, our goal is to correctly predict $\mathcal{Y}$. We assume we are given a cost for misclassifying a node;

when we classify a node as y_k whereas the correct assignment is y_l, we incur a cost of c_{kl}. The expected misclassification cost (EMC) for one node is then given by:

$$EMC(Y_i|\mathcal{X}=x, CM) = \min_{y_k} \sum_{y_l \neq y_k} P(Y_i = y_l|\mathcal{X}=x, CM) \times c_{kl}$$

We are trying to find the joint assignment to the $\mathcal{Y}$ that minimizes the total expected misclassification cost,

$$\sum_{Y_i \in \mathcal{Y}} EMC(Y_i|\mathcal{X}=x, CM).$$

2.2 Label Acquisition

As mentioned in the introduction, we are interested in settings where we are able to ask for the labels for some of the nodes. More formally, we consider the case where we can acquire the values for a subset of the labels $\mathcal{A} \subseteq \mathcal{Y}$. The total misclassification cost then changes as follows:

$$\sum_{Y_i \in \mathcal{Y} \setminus \mathcal{A}} EMC(Y_i|\mathcal{X}=x, \mathcal{A}, CM)$$

However, we do not know the values of $\mathcal{A}$ before we acquire them. Thus, we take an expectation over possible the values.

$$\sum_{Y_i \in \mathcal{Y} \setminus \mathcal{A}} \sum_{a} P(\mathcal{A}=a) EMC(Y_i|\mathcal{X}=x, \mathcal{A}=a, CM)$$

Of course, acquiring the value of a label is costly. Let the cost of acquiring value of the label Y_i be C_i. Extending it to sets, $C(\mathcal{A}) = \sum_{Y_i \in \mathcal{A}} C_i$. Then, the total cost we incur, and thus the objective function is the sum of the expected misclassification cost and the acquisition cost:

$$L(\mathcal{A}) = \sum_{Y_i \in \mathcal{Y} \setminus \mathcal{A}} \sum_{a} P(\mathcal{A}=a) EMC(Y_i|\mathcal{X}=x, \mathcal{A}=a, CM) + C(\mathcal{A})$$

Given a spending budget B, the label acquisition problem is then to find the optimal subset

$$\mathcal{A}^* = \operatorname*{argmin}_{\mathcal{A} \subseteq \mathcal{Y}, C(\mathcal{A}) \leq B} L(\mathcal{A})$$

minimizing the sum of expected misclassification cost and acquisition cost.

The computational complexity of finding the optimal $\mathcal{A}^*$ depends on two main factors. The first is the computational complexity of computing and updating the conditional probabilities $P(Y_i|\mathcal{X}=x, \mathcal{A}=a)$ for the expected misclassification computations. The second factor is how the search space of $\mathcal{A}^*$ is factored when a specific Y_i is acquired. Unless very restrictive assumptions about the structure of the underlying collective model are made, such as linear dependence on the neighborhood and attributes, the problem is at least $\mathbf{NP^{PP}}$-complete [10], and this is the case for our model CM.

Since finding the optimal solution to the label acquisition problem is intractable in a general setting, we must resort to approximate techniques. In the next sections, we describe three such techniques. The first is a greedy strategy based on approximate inference. The second is based on an analogy to viral marketing. The third is a simple yet effective and intuitive approach based on learning how to correct the mistakes. We also compare to simple strategies based on structural properties of the network.

3. APPROXIMATE INFERENCE AND GREEDY ACQUISITION (AIGA)

There are two reasons why finding the optimal solution is intractable: 1) exact probability computation is intractable in general, and 2) unless the probability space for $\mathcal{A}^*$ can be factored, one needs to consider all possible acquisition subsets $\mathcal{A} \subseteq \mathcal{Y}$, which is exponential in the size of $\mathcal{Y}$.

The first technique we propose, approximate inference and greedy acquisition (AIGA), is the most obvious approach. Instead of computing the exact probabilities when calculating the expected misclassification costs, we approximate them. For instance, when the underlying model is an undirected graphical model, such as Markov random field, there are many approximate inference techniques we can make use of, such as loopy belief propagation [28], variational methods [8], Gibbs sampling [5], etc. If the underlying model is a collection of local conditional models, then one can use iterative approaches to compute the probabilities [19, 13].

Moreover, instead of considering all possible subsets of $\mathcal{Y}$, we take a greedy selection approach, where we consider each label independently and greedily acquire the label that minimizes the objective $L(\mathcal{A})$.

Let the $value_{aiga}(Y_i)$ be defined as the reduction in the objective function when the value of Y_i is acquired:

$$value_{aiga}(Y_i) \triangleq L(\mathcal{A} \cup \{Y_i\}) - L(\mathcal{A}).$$

The overall label acquisition problem is then solved by using approximate inference to find the label Y_i that has the maximum $value(Y_i)$ and whose inclusion in the acquisition set is not going to cause us to exceed the budget, acquire the value for it, and repeat the process until the budget is exhausted (Algorithm 1). Note that the *value* calculation at step 7 is essentially an *expected value of information* calculation [6] and it requires running the inference procedure for each possible label.

Algorithm 1: Approximate inference and greedy acquisition (AIGA) algorithm.

```
  Input: $G$ – the test graph, $CM$ – the learned collective
         model, $c_{ij}$ – misclassification costs, $C_i$ – the
         acquisition costs, $B$ – the budget
  Output: $\mathcal{A}$ – the set of acquisitions
1 $\mathcal{A} \leftarrow \emptyset$
2 while $C(\mathcal{A}) < B$ do
3     $Y_{max} \leftarrow nil$
4     $maxValue \leftarrow -\infty$
5     for $Y_i \in \mathcal{Y} \setminus \mathcal{A}$ do
6         if $value_{aiga}(Y_i) > maxValue \wedge C(\mathcal{A} \cup \{Y_i\}) \leq B$
          then
7             $maxValue \leftarrow value_{aiga}(Y_i)$
8             $Y_{max} \leftarrow Y_i$
9     $\mathcal{A} \leftarrow \mathcal{A} \cup \{Y_{max}\}$
```

The success of this method depends on a number of things. First, the accuracy of this method depends heavily on the precision of the estimated probability values. If the probability estimates are not well-calibrated, then the expected misclassification costs will be incorrect [29], making the $value_{aiga}$ calculations meaningless. Second, the time this acquisition method takes to run depends on the time complexity of the approximate inference technique; we need to calculate the $value_{aiga}$ of each label, which requires running the inference algorithm once for each node and once for each possible value of the label. Thus, when the number of nodes and the number of possible labels are high, and if the inference technique is expensive, this acquisition method can be very slow.

4. VIRAL MARKETING ACQUISITION (VMA)

Another, simpler, approach to label acquisition is based on an analogy to viral marketing [9, 23]. In the viral marketing setting, we have customers that are potential buyers of a product and the customers have relationships between each other, such as family, friendship, co-worker, etc. When a customer buys a product, the customer advertises it (by word of mouth) to his or her neighbors in the network. Through marketing, we can (hopefully) increase the chance that a customer will buy a product by marketing to the right set of customers. Thus, similar to the label acquisition problem, there is then a question of to which subset of customers we should market, in the hope that these customers will like the product, buy it, and recommend it to their neighbors, who will hopefully buy and recommend it in turn.

The analogous mapping to label acquisition for collective classification is as follows. There are nodes (customers) that we need to classify and we have the choice to acquire the labels for (market to) some of them. Our task is to choose an initial set of labels to acquire so that the number of correctly classified nodes (the customers who buy the product) in the end is maximized. This implicitly assumes that the misclassification costs c_{ij} are symmetric and equal; i.e., $c_{ij} = c_{ji}$ for all i, j.

There are many viral marketing approaches that differ in the formulation of the problem, the assumptions that they make, and the solutions that they offer [9, 23]. Reviewing these work and discussing the differences is beyond the scope of this paper. Our viral marketing formulation is based on one of the recent formulations, the formulation of Richardson & Domingos [23], that has an exact solution. We next describe the details of the formulation and the mapping.

For the viral marketing formulation, for each node V_i, we introduce a new indicator variable T_i, which indicates whether Y_i is predicted correctly. Whether a prediction is correct depends on the informativeness of the attributes X_i, whether its neighbors $\mathcal{N}_i$ are correctly classified, and which labels are acquired, $\mathcal{A}$. Following [23] we make the assumption that this probability is a linear combination of a local probability and a relational probability as follows:

$$P(T_i|\mathcal{N}_i, X_i, \mathcal{A}) \triangleq \beta_i P_l(T_i|X_i, \mathcal{A}) + (1-\beta_i) P_r(T_i|\mathcal{N}_i, \mathcal{A})$$

where β_i denotes how much an instance depends on its local attributes versus its neighbors, where the local probability P_l is defined as:

$$P_l(T_i|X_i, \mathcal{A}) \triangleq \begin{cases} 1 & \text{if } Y_i \in \mathcal{A} \\ \max\limits_{y_k} P(Y_i = y_k|X_i) & \text{otherwise} \end{cases}$$

and the relational probability P_r is a linear combination of the statuses of the neighbors:

$$P_r(T_i|\mathcal{N}_i, \mathcal{A}) = \frac{1}{|\mathcal{N}_i|} \sum_{Y_j \in \mathcal{N}_i} T_j.$$

The probability $P(Y_i = y_k|X_i)$ can be computed by learning a classifier on the train graph G^{tr}.

The objective now is to make acquisitions so as to maximize the total probability of correctly classifying the nodes in the network. To find out which labels will be the most valuable ones, we calculate two intuitive measures. The first one measures how much a unit change in $P_l(T_i|X_i, \mathcal{A})$ will affect the network:

$$\Delta(Y_i) \triangleq \sum_{V_j \in \mathcal{V}} \frac{\partial P(T_j = 1|X_j, \mathcal{A})}{\partial P_l(T_i|X_i, \mathcal{A})}$$

The second one measures how much an instance's probability of correct classification is increased when we acquire the label for it:

$$\Delta P(Y_i) = \beta_i \left(P_l(T_i|X_i, \mathcal{A} \cup Y_i) - P_l(T_i|X_i, \mathcal{A}) \right)$$

Then, the effect that acquiring a label Y_i will have in the network, i.e., the value of a label is just a product of the two.

$$value_{vma}(Y_i) = \Delta(Y_i) \Delta P(Y_i)$$

We omit some of the details about how to derive these equations. The interested reader can refer to [23].

The acquisition strategy is then as follows. First compute the $value_{vma}$ of each label, and then iteratively acquire the label with the highest value until the budget is exhausted (Algorithm 2). Note that because this particular viral marketing model has an exact solution, the values for the labels need not be recomputed at each step.

Algorithm 2: Viral marketing based acquisition (VMA) algorithm. Assumes uniform costs for the labels, and assumes the misclassification costs are symmetric.

```
Input: $G$ – the test graph, $C_i$ – the acquisition costs
       (assumed uniform), $B$ – the budget
Output: $\mathcal{A}$ – the set of acquisitions
$\mathcal{A} \leftarrow \emptyset$
while $C(\mathcal{A}) < B$ do
    $Y_{max} \leftarrow nil$
    $maxValue \leftarrow -\infty$
    for $Y_i \in \mathcal{Y} \setminus \mathcal{A}$ do
        if $value_{vma}(Y_i) > maxValue \wedge C(\mathcal{A} \cup \{Y_i\}) \leq B$
        then
            $maxValue \leftarrow value_{vma}(Y_i)$
            $Y_{max} \leftarrow Y_i$
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{Y_{max}\}$
```

With these assumptions, our formulation is same as that of [23] with only one subtle difference. In the viral marketing domain, when a person is marketed a product, there is still a non-zero probability for that person not buying the product. In label acquisition, however, we assume that we can acquire labels with perfect information; that is, there is no uncertainty about a node's label after we acquire it.

5. REFLECT AND CORRECT (RAC)

The next method that we introduce is based on a simple intuition: The set of instances that the collective classification model misclassifies tend to be clustered together because misclassifying one instance makes it very likely that its neighbors will be misclassified as well (propagation of incorrect information). Thus, there are islands (or peninsulas) of misclassification in the graph – sets of connected nodes that are misclassified. If we can find these islands of misclassification, then we can potentially trigger correct classification of those islands by acquiring labels for a few of the nodes in these islands. The question is then how to find the islands of misclassification.

We first focus on finding out when a prediction for a particular node is correct. We again associate a random variable T_i with each $V_i \in \mathcal{V}$, denoting whether the prediction for Y_i was indeed correct. But, this time, instead of using the modeling we did in the viral marketing approach, we construct some features that are possible indicators of whether a node is misclassified, and we learn a classifier on these features to model the dependence of T_i on the constructed features. To perform the learning phase, we use the label information of the training graph G^{tr}, and predictions of the collective model CM on the training graph. We next describe the features we constructed for this task.

We construct three simple features, one local, one relational, and one global. Intuitively, the local feature captures how much the attributes disagree with the classification decisions of the collective model CM. The relational feature captures how likely it is that the neighbors of an instance are also misclassified. Lastly, the global feature captures how different the posterior distribution of the classes is from the expected prior distribution. We next explain these features in detail and provide mathematical definitions for them.

The *local feature* measures how far the prediction of CM is from the truth according to the attributes. Assume that we predict $Y_i = y_j$ using CM. Then, the local feature for node V_i is defined as:

$$lf_i \triangleq 1 - P(Y_i = y_j | X_i)$$

Again, we can compute $P(Y_i = y_j|X_i)$ by learning a local classifier on the nodes of the train graph G^{tr}. The intuition behind the local feature lf is that if the attributes of a node disagree with the prediction based on CM, then it is a signal for a possible misclassification. The local feature is a measure of the strength of the disagreement between the local classifier and the collective model.

The *relational feature* captures how likely that a node's neighbors are also misclassified. The intuition is that if a node's neighbors are misclassified, then the node itself is probably misclassified as well (because the model is a collective model). There are different possibilities for defining the relational feature; for instance, it can be defined as a recursive function of T_i, and then it can be computed iteratively. We take the simplest approach and define it as the average of the local features, lf_j, of the neighbors of the node V_i.

$$rf_i \triangleq \frac{1}{|\mathcal{N}_i|} \sum_{Y_j \in \mathcal{N}_i} lf_j$$

Lastly, the *global feature* captures the difference between our prior belief about the class distributions and the posterior distribution that we get based on the predictions. For example, based on our prior belief, if we expect to classify 20% of the nodes with label y_j, but CM predicts 60% of the nodes as label y_j, then some of the nodes that are classified as y_j are probably misclassified.

Let the prior distribution of the class y_j be denoted by $Prior(y_j)$ and let the posterior distribution based on the predictions of CM be denoted by $Posterior(y_j)$. The $Prior(y_j)$ can be estimated from the training graph G^{tr}. Then, we define the global feature for the node V_i that is predicted as y_j as follows:

$$gf_i \triangleq \frac{Posterior(y_j) - Prior(y_j)}{1 - Prior(y_j)}$$

Having constructed these three features, we learn a classifier for $P(T_i|lf_i, rf_i, gf_i)$. The training data for this classifier comes from the training graph G^{tr} and the predictions of CM on this training graph. We used logistic regression but any vector based classifier will work. Next, we define the value of a particular acquisition.

$$\begin{aligned} value_{rac}(Y_i) = \quad & \delta(P(T_i = 0|lf_i, rf_i, gf_i) > 0.5) + \\ & \sum_{Y_j \in \mathcal{N}_i} \delta(P(T_j = 0|lf_j, rf_j, gf_j) > 0.5) \end{aligned}$$

where $\delta(predicate) = 1$ if the *predicate* is true, 0 otherwise. The value of acquiring the label Y_i is the number of nodes that are misclassified in the neighborhood of this label, including itself, that can potentially be corrected by this acquisition. Our acquisition method is then to greedily acquire the labels that have the highest value, until the budget is exhausted (Algorithm 3). We again assume symmetric misclassification costs and uniform acquisition costs.

Algorithm 3: Reflect and Correct (RAC) based acquisition algorithm. Assumes uniform costs for the labels, and assumes the misclassification costs are symmetric.

```
Input: $G$ – the test graph, $C_i$ – the acquisition costs
       (assumed uniform), $B$ – the budget
Output: $\mathcal{A}$ – the set of acquisitions
$\mathcal{A} \leftarrow \emptyset$
while $C(\mathcal{A}) < B$ do
    $Y_{max} \leftarrow nil$
    $maxValue \leftarrow -\infty$
    for $Y_i \in \mathcal{Y} \setminus \mathcal{A}$ do
        if $value_{rac}(Y_i) > maxValue \wedge C(\mathcal{A} \cup \{Y_i\}) \leq B$
        then
            $maxValue \leftarrow value_{rac}(Y_i)$
            $Y_{max} \leftarrow Y_i$
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{Y_{max}\}$
    Update the predictions on $\mathcal{Y} \setminus \mathcal{A}$ using the acquired
    value of $Y_{max}$ and $CM$.
    Update the constructed features based on the new
    predictions.
```

We next describe the experimental setup and results on some synthetic and real datasets.

6. EXPERIMENTS

We compared the three methods, AIGA, VMA, and RAC, two methods that are based on network structural measures, a random (RND) acquisition method, no acquisition (NONE), and perfect information PI, and we report accuracies on both real and synthetic datasets.

The first of the acquisition methods based on network structure is *degree acquisition*, (DEG); it first ranks the nodes according to their degree in the graph and then acquires labels until the budget is exhausted. The second method, *K-Mediods clustering* (KM), first clusters the network into a prespecified number of clusters and then acquires the labels for the centers of the clusters. Rattigan et al. [22] showed that KM outperformed other structural methods, such as betweenness, degree, closeness, etc.

The accuracies corresponding to no acquisition, NONE, were useful in two ways. When this method performs poorly, it is a good indicator for possible "floods" in a graph. We are also able to see how much the acquisitions helped. The perfect information PI accuracies, on the other hand, are useful to compare how close the acquisition methods are to the optimal solution. Recall that finding the optimal solution is intractable; thus, we computed PI accuracies by letting the collective classifier look at the labels of the neighbors of a node when it is making a decision about the node. Note that PI accuracy can still be suboptimal because PI is evaluated on $\mathcal{Y}$ whereas the acquisition methods are evaluated on $\mathcal{Y}\backslash\mathcal{A}$. $\mathcal{A}$ can potentially include the noisy labels on which even the Bayes Optimal classifier can make mistakes.

To be able to compare all these methods, we assumed uniform acquisition costs and symmetric misclassification costs. Thus, the budget B determined how many labels we can acquire and we used accuracies to compare the methods with each other. For both synthetic and real datasets, we used a pairwise Markov random field [26] as our collective model CM and used loopy belief propagation [28] for the inference. As a local model, LM, of the attributes, we used Naive Bayes, and we used Logistic Regression for RAC. We used Naive Bayes for the local attributes because that matched the generative model for the synthetic data. Naive Bayes also performed comparably well with logistic regression on the real data. We used logistic regression for RAC because the features were numeric and logistic regression was able to handle them better than Naive Bayes with Gaussians.

6.1 Experiments on Synthetic Data

We generated synthetic data using the forest-fire graph generation model [12]. The forest fire model is shown to exhibit many real-world phenomenon such as power law degree distribution, small world effect, and shrinking diameters. However, the forest-fire method, like most random network generators, does not generate labels and attributes for the nodes. In order to label the nodes, we used the method that Rattigan et al. [22] described, and after generating the labels for the nodes, we generated attributes for each node using a Naive Bayes model.

For our evaluation, for each training graph, we learned our collective, local, and RAC models, and then generated five test graphs and compared the acquisition methods on the test graphs, varying the number of labels acquired. We repeated this procedure five times. We report average accuracies over the 25 test graphs.

6.1.1 Repeatability

Following Rattigan et al. [22], we used a forward burning probability of 0.4 and a backward burning probability of 0.2. We labeled each node with one of 5 possible labels and generated 20 binary attributes using a simple Naive Bayes generation model; 4 attributes - indexed by the class - were generated with a probability of success of 0.65 for the correct label and 16 attributes were generated with a probability of success of 0.4 for the other labels. We varied the number of nodes for different experiments and we report those numbers in the respective results sections. We used $\beta = 0.5$ for the VMA approach, following [23].

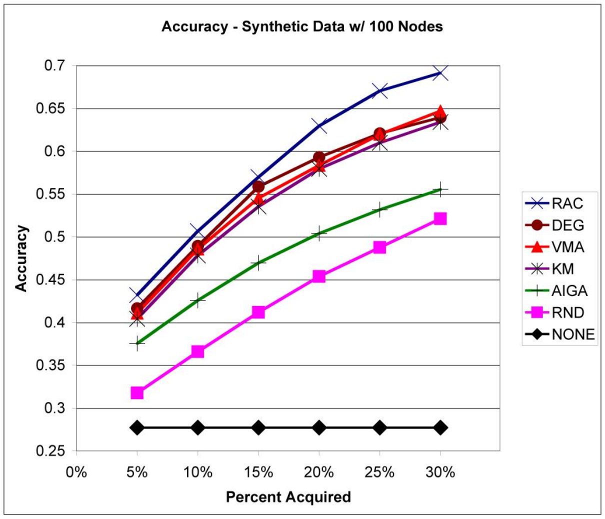

Figure 1: Accuracy comparison for graphs of 100 nodes. All methods significantly outperform the random acquisition. AIGA does worse than other acquisition methods. There are not significant differences between VMA, DEG, and KM. RAC outperforms all methods, and the differences are statistically significant starting with 20% acquisition.

6.1.2 Results

Even though the AIGA method is a polynomial-time algorithm, each single acquisition decision requires running inference for each node and for each possible value of its label. Thus, it is impractical to run AIGA on large graphs. We begin by comparing all the methods including AIGA on small graphs, graphs of 100 nodes, and then compare the remaining methods on larger graphs of 2000 nodes.

For each experiment, we first report the average degree, assortativity [20], how well the local model LM does on average, and the average perfect information PI accuracies.

The first set of graphs of 100 nodes had an average degree of 3.36 and an assortativity of 0.62. LM had an average accuracy of 0.62, NONE had an average accuracy of 0.28, and average PI accuracy was 0.80. These large differences between NONE and LM and NONE and PI are a good indication of "flooding." The big difference between LM and PI also shows that collective classification has the potential to improve dramatically over flat classification. We varied the percentage of labels acquired from 5% to 30% with 5% increments. We show the accuracy comparisons for the acquisition methods in Figure 1.

There are four important results to observe from Figure 1. The first one is that label acquisition can alleviate flooding and that the choice of which nodes to label does matter because all informed methods outperformed the random acquisition significantly at all levels of acquisition.

Second, all other informed acquisition methods outper-

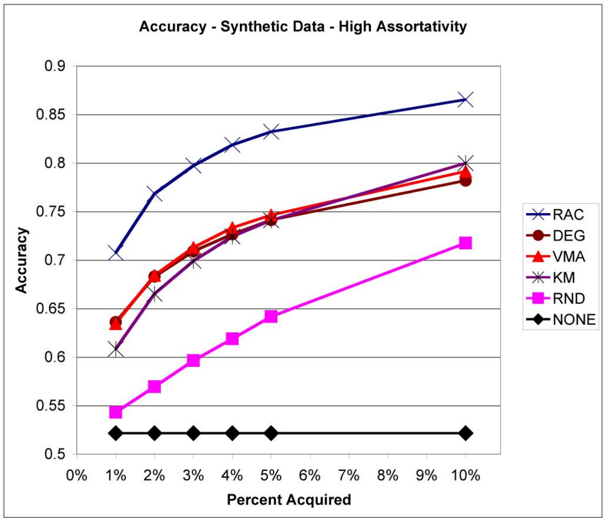

Figure 2: Accuracy comparison for graphs of 2000 nodes and high assortativity. RAC significantly outperforms all other methods at all percentages. The differences between RAC and the closest runner-up is sometimes 10% i.e., 200 nodes. The differences between VMA, DEG, and KM are not statistically significant.

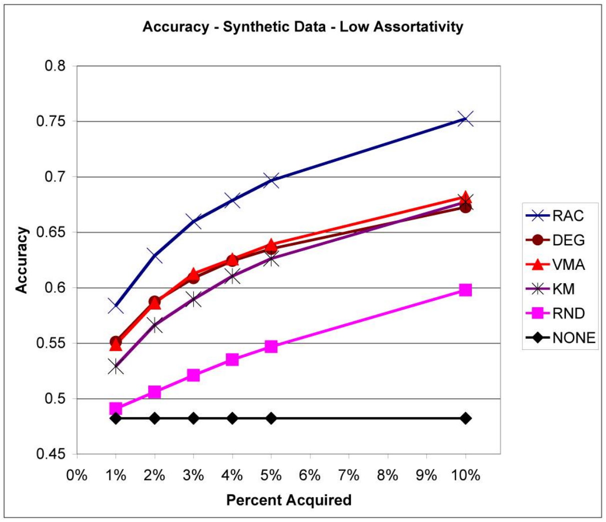

Figure 3: Accuracy comparison for graphs with low assortativity. We observe similar trends with the previous experiments. The only difference is that the accuracy gaps are not as big anymore, because nodes have a lesser impact on their neighbors due to lower assortativity.

formed the AIGA method significantly. This is surprising at first, because one would expect the AIGA method to perform the best. However, recall that the inference technique is loopy belief propagation, which is an approximate method of probability computation, and it is known to produce suboptimal results when there are many short cycles in the graph. Given the assortativity of the data, we see that beliefs about the nodes' labels reinforce each other iteratively, and thus most of the probability distributions for the nodes' labels are spike distributions that spike at value 1 for one label value and at 0 for the other values. Because the probabilities are extreme, the $value_{aiga}$ computations for labels were not very discriminative.

The third observation is that KM did not perform better than the DEG method. This is in contrast to the results that Rattigan et al. [22] observed. There are at least three possible explanations for DEG performing equally well, or sometimes better than KM. The first reason is that we use attributes in our setup whereas in their setup only node labels were used. The second reason is our collective model is a pairwise MRF whereas they used a relational neighbor classifier [14]. And a third reason may be that the DEG heuristic is breaking cycles and thus allowing the loopy belief propagation to converge to the correct distribution.

The final observation is that the RAC method outperformed all the other methods at all levels of acquisition. The differences between RAC and AIGA and RAC and KM are statistically significant at all percentages. The difference between RAC and VMA and RAC and DEG became statistically significant at percentage 20% and remained significant thereafter.

We next compare the acquisition methods except AIGA on much larger graphs, graphs of 2000 nodes. We varied the level of assortativity to compare the methods at different settings. We first discuss results for high assortativity.

The average degree for highly assortative graphs was 3.83 and the mean assortativity was 0.80. LM had an average accuracy of 0.51 and PI accuracy as 0.93. We varied the percentage of labels acquired from 1% to 5% with 1% increments, and we also show comparison at 10% to show the longer term trends. We show the accuracy comparisons for the acquisition methods in Figure 2.

We observe similar trends with the experiments on the larger graphs, only at a finer detail. Flooding is a problem that needs to be dealt with and RAC outperforms all other methods significantly at all percentages. The difference between RAC and the closest runner-ups are sometimes as high as 10% (i.e., 200 more nodes are labeled correctly by RAC), and the difference between RAC and random acquisition is sometimes as high as 20%. As for VMA and DEG, there is not a clear winner. Both approaches outperform KM initially, but KM catches up, and even outperforms them when we acquire 10% of the labels, though the differences are not statistically significant at $p = 0.05$.

It is important to note that the forest fire model generates graphs with power-law degree distributions. That is, while there are some high degree nodes, there are not that many high degree nodes in graph. Thus, after a certain number of labels are acquired by the DEG method, the labels corresponding to the high degree nodes will already be acquired, and DEG will be choosing any regular node. When that happens, KM becomes a more intelligent method.

We finally present results on graphs with lower assortativity. The average degree was 3.78 and the mean assortativity was 0.61. LM had an average accuracy of 0.51 and PI accuracy as 0.85. We show the accuracy comparisons in Figure 3.

We again observe similar trends; the only difference is that the improvement over random acquisition is not quite as pronounced because acquiring a label for a node does not affect its neighbors as much anymore due to lower assortativity.

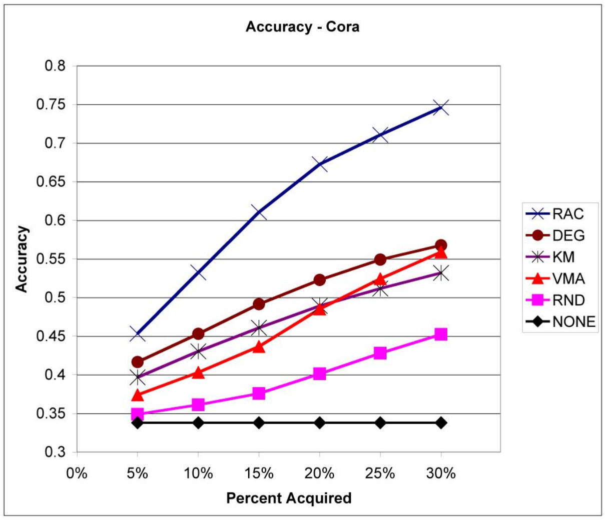

Figure 4: Accuracy comparison for the Cora dataset. The RAC method performed significantly better the other methods, a difference of up to 18% compared to the closest runner-up. The differences between the other methods were not statistically significant.

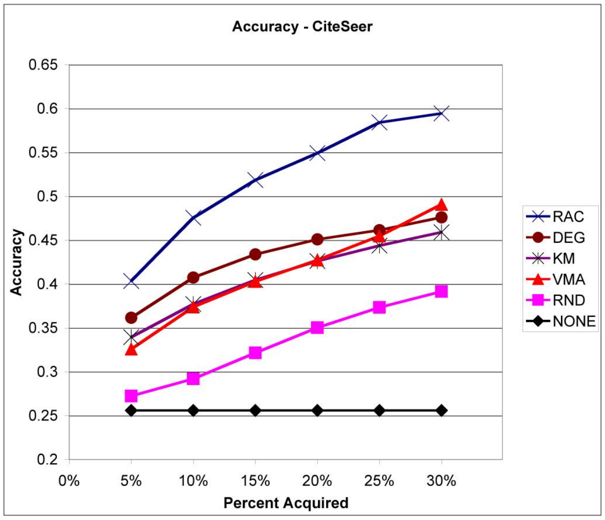

Figure 5: Accuracy comparison for the CiteSeer dataset. RAC outperformed all other methods significantly. The differences between the other methods were not statistically significant.

6.2 Experiments on Real Data

We experimented on two real publication datasets that are publicly available, the Cora dataset [17] and the CiteSeer dataset [4]. The Cora dataset contains a 2708 machine learning papers that are divided into 7 classes, while CiteSeer dataset has 3312 documents that are divided into 6 classes.

Our evaluation methodology is slightly different from the general practice. In real life scenarios, we usually have much smaller labeled data compared to the unlabeled data. This difference in the proportion makes the interactions between the unlabeled nodes more common than the interactions between the labeled and unlabeled nodes. To mimic these two observations, we adopted the following evaluation strategy.

We divided each dataset into three disjoint splits and repeatedly trained on one split and tested on the remaining two (in contrast to training on two splits and testing on the other). Additionally, we did not make use of the edges between the labeled nodes and the unlabeled ones during inference.

Because of these changes in the evaluation strategy, which we believe results in a more realistic evaluation, the accuracies corresponding to NONE are very low compared to the numbers reported in the literature. The primary reason is that the test graph is more amenable to flooding now, because it is large and there are no interactions between the test graph and the training graph. However, the PI accuracies are close to the previously reported numbers [24].

We first present results on the Cora dataset. The Cora splits had an average degree of 3.32 and mean assortativity of 0.79, which is relatively high. This lead to a noticeable difference between the LM performance and PI performance, which were 0.61 and 0.77 respectively. The accuracy comparisons for the Cora dataset are presented in Figure 4.

We observed similar trends in Cora to the ones we observed in the synthetic datasets. RAC performed significantly better than other methods, achieving up to 18% accuracy difference over the closest runner-up, which is the DEG method. The DEG method outperformed both VMA and KM but the differences were not statistically significant. Interestingly, VMA started with a low accuracy compared to others but improved more steeply.

We finally present results on the CiteSeer dataset. The splits had an average degree of 2.71 and a mean assortativity of 0.68, which was lower compared to that of Cora, thus the difference between LM and PI was not as large; LM performance was 0.57 and PI performance was 0.61. However, even with lower assortativity, we observed the "island effect," the accuracies without any acquisition was considerably low (Figure 5). For this dataset, we observed exactly the same trends we had for the Cora dataset.

We ran some preliminary experiments using Iterative Classification Algorithm [13] instead of a pairwise MRF as the underlying collective model and observed similar results. We omit the results due to space limitations.

7. RELATED WORK

Substantial research has been done on the area of active learning [3, 27, 16]. While active learning is related to label acquisition during inference, the aim for active learning is to acquire labels to learn a good model. We are instead acquiring labels during inference.

Another related area is viral (or targeted) marketing [9, 11, 21, 23], where a subset of customers need to be selected for targeted advertisement so as to maximize the product sales. We showed how viral marketing is related to label acquisition and used Richardson & Domingos's model [23] to compare against. Other models could very well be used and compared against; one of the reasons we chose [23] is the fact that the exact solution was tractable.

The work in feature-value acquisition during testing [1, 25] is very related to the label acquisition problem; however, the focus has been on acquiring feature values, not labels, and also they considered acquisition for non-graph data.

The most related work is that of Rattigan et al. [22]. As far as we know, they are the first to publish directly on label acquisition during inference. They compared different acquisition methods based on network structural measures, such as degree and betweenness, and they suggested a method based on clustering the network and empirically showed that the clustering method performed the best. They assumed that the nodes did not have any attributes, thus their method did not require any training data. We made different assumptions about the data, that is, the nodes had attributes and we had some training data available. We implemented their method and compared with it.

Finally, cautious inference [18, 19] can be used to alleviate the problem of propagation of incorrect information. We explored using cautious inference, and it helped when the islands of misclassifications were small, but it did not solve the problem when the majority of the network was flooded with a small number of labels.

8. LIMITATIONS AND FUTURE WORK

One of the limitations of the RAC method is that it is based on the assumptions that the misclassification costs are symmetric and the acquisition costs are uniform. The second assumption can be lifted by making use of the probabilities that the RAC classifier produces about whether a node is misclassified. However, lifting the first assumption requires further research.

The RAC method can very well be applied to the viral marketing domain. RAC can be trained on some labeled data and can be used to find out "islands of non-buyers," as we used it to find islands of misclassification. Then, targeted advertisement can be done to those islands.

9. CONCLUSIONS

We formulated the problem of label acquisition during inference and discussed why it is an important and hard problem. We described three informed methods and compared them to two methods that are based on network structural measures. The first method that we described is the most straightforward one and is based on approximate inference and greedy acquisition. We showed that it does not perform well in practice. We described another method that is a direct application of viral marketing to label acquisition. This method performed equally well with the network structural methods. Finally, we proposed a method that is based on learning when a collective model makes mistakes and suggests acquisitions to correct those mistakes. We empirically showed that this method significantly outperformed all other methods on both real and synthetic datasets.

Acknowledgments: This work was supported by the National Science Foundation, NSF #0423845 and NSF #0438866, with additional support from ARO #W911NF0710428.

10. REFERENCES

[1] M. Bilgic and L. Getoor. VOILA: Efficient feature-value acquisition for classification. In *AAAI*, 2007.

[2] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *SIGMOD*, 1998.

[3] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[4] C. L. Giles, K. D. Bollacker, and S. Lawrence. Citeseer: an automatic citation indexing system. In *Proc. of ACM conf. on Digital Libraries*, 1998.

[5] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Interdisciplinary Statistics. Chapman & Hall/CRC, 1996.

[6] R. A. Howard. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, 2(1):22–26, 1966.

[7] D. Jensen, J. Neville, and B. Gallagher. Why collective inference improves relational classification. In *KDD*, 2004.

[8] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 1999.

[9] D. Kempe, J. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.

[10] A. Krause and C. Guestrin. Optimal nonmyopic value of information in graphical models - efficient algorithms and theoretical limits. In *IJCAI*, 2005.

[11] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Transactions on the Web*, 1(1):5, 2007.

[12] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM TKDD*, 1(1):177–187, 2007.

[13] Q. Lu and L. Getoor. Link based classification. In *ICML*, 2003.

[14] S. Macskassy and F. Provost. A simple relational classifier. In *Workshop on Multi-Relational Data Mining in conj. with KDD (MRDM)*, 2003.

[15] S. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *Journal of Machine Learning Research*, 8:935–983, 2007.

[16] A. McCallum and K. Nigam. Employing em and pool-based active learning for text classification. In *ICML*, 1998.

[17] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Inf. Retrieval*, 3(2):127–163, 2000.

[18] L. McDowell, K. M. Gupta, and D. W. Aha. Cautious inference in collective classification. In *AAAI*, 2007.

[19] J. Neville and D. Jensen. Iterative classification in relational data. In *SRL Workshop in AAAI*, 2000.

[20] M. E. J. Newman. Mixing patterns in networks. *Physical Review E*, 67(2):026126, Feb 2003.

[21] F. Provost, P. Melville, and M. Saar-Tsechansky. Data acquisition and cost-effective predictive modeling: targeting offers for electronic commerce. In *Proc. of Int. Conf. on Electronic Commerce*, 2007.

[22] M. Rattigan, M. Maier, and D. Jensen. Exploiting network structure for active inference in collective classification. In *ICDM Workshop on Mining Graphs and Complex Structures (MGCS)*, 2007.

[23] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD*, 2002.

[24] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. Technical Report CS-TR-4905, University of Maryland, College Park, 2008.

[25] V. S. Sheng and C. X. Ling. Feature value acquisition in testing: a sequential batch test algorithm. In *ICML*, 2006.

[26] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *UAI*, 2002.

[27] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2002.

[28] J. Yedidia, W.T.Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS*, 2000.

[29] B. Zadrozny and C. Elkan. Learning and making decisions when costs and probabilities are both unknown. In *KDD*, 2001.