Query-driven Active Surveying for Collective Classification

Galileo Namata, Ben London, Lise Getoor, Bert Huang {NAMATAG,BLONDON,GETOOR,BERT}@CS.UMD.EDU
Department of Computer Science, University of Maryland, College Park, MD 20742

Abstract

In network classification problems such as those found in intelligence gathering, public health, and viral marketing, one is often only interested in inferring the labels of a subset of the nodes. We refer to this subset as the *query set*, and define the problem as *query-driven collective classification*. We study this problem in a practical *active learning* framework, in which the learning algorithm can *survey* non-query nodes to obtain their labels and network structure. We derive a surveying strategy aimed toward optimal inference on the query set. Considering both feature and structural smoothness, concepts that we formally define, we develop an algorithm which adaptively selects survey nodes by estimating which form of smoothness is most appropriate. We evaluate our algorithm on several network datasets and demonstrate its improvements over standard active learning methods.

1. Introduction

Collective classification, the task of labeling nodes in a network, is an important problem in many domains, such as analysis of social networks, biological networks, and citation databases (Macskassy & Provost, 2007; Sen et al., 2008). While traditional learning aims to learn a predictor for all available data, we consider the case in which one is primarily interested in labeling a particular subset of nodes, which we refer to as the *query set*. For example, when labeling a social network, we may only be interested in the labels of key high-ranking or influential individuals; classifying the rest of the network may only be desired to aid in collectively classifying the targeted nodes. We refer to this problem as *query-driven collective classification*.

In many practical scenarios, labels and network structure may not be immediately available for all nodes, and certainly are not available for the nodes in the query set. Instead, there is a cost for acquiring this information. We therefore explore the problem of query-driven collective classification in an active learning setting. In traditional active learning, the learner controls the sequence of training examples received. Unlike previous work (Bilgic et al., 2010; Kuwadekar & Neville, 2011; Macskassy, 2009; Settles, 2009), we do not restrict the training examples to simple instance-label pairs; we instead explicitly consider other information that is inherent to relational domains. This leads to a more general view of information acquisition, which we refer to as *active surveying*. Whereas prior work in active surveying (Sharara et al., 2011) was geared specifically to the problem of identifying opinion leaders, here we present a more general view. In our setting, a survey returns not only the label(s) of a node, but also any missing links incident on that node (i.e., the node's ego network). Our work is also related to work in *active sampling* (Pfeiffer III et al., 2012) which similarly acquires both label and edge information but for the distinct task of discovering all nodes with a specific label value.

We assume that the algorithm cannot directly survey a query node. For various reasons in practice, surveying a query node may incur a prohibitive cost (e.g., query nodes may be uncooperative or unreachable). Thus, the challenge is to identify the optimal subset of non-query nodes to survey, subject to budget constraints, that will enable us to correctly predict the labels of the query set.

We analyze the surveying problem using a distributional "smoothness" assumption, where we define a query-driven problem to be smooth if the distribution of labels, conditioned on some measurable distance function, changes proportionally to the distance. This distance function can be computed using features or network structure, depending on the problem domain. If the smoothness property holds for a given dataset and metric, then surveying nodes based on their proximity to the query nodes should minimize the deviation between the query and survey node distributions. Therefore, the smoothness assumption theoretically implies that minimizing this distance thereby minimizes the average loss over the query set. Based on this analysis, we develop several active surveying strategies: one that leverages feature smoothness; one that leverages structural smoothness; and a novel adaptive algorithm that automatically chooses between the two, based on an empirical estimate of the so-called *assortativity* in the current observed graph. We evaluate these strategies on several real-world networks using an iterative classification algorithm to perform collective classification.

Appearing in *Proceedings of the Workshop on Mining and Learning with Graphs (MLG-2012)*, Edinburgh, Scotland, UK, 2012.

2. Motivating Examples

In this section, we present three real-world examples of active surveying for query-driven collective classification.

Intelligence Gathering The query-driven active setting is particularly apt for intelligence gathering, specifically for analyzing organized crime and terrorist networks. In this scenario, we may be interested in ascertaining the affiliation, disposition, or role (i.e., label) of key individuals (i.e., query nodes) in a population. For context specific reasons, these individuals may be inaccessible, making it difficult, if not impossible, to acquire this information directly. Moreover, the full network may be largely unobserved. Through surveillance, we can acquire information about the network, including the labels of less important people, who may be more accessible. Surveillance or investigation, however, are expensive in terms of both time and resources, and so we aim to identify the optimal set of people to investigate, given a budget.

Disease Transmission Consider the task of monitoring the spread of an infectious disease in a partially observed, potentially noisy social network. In this context, the goal is to determine the infection status of "at-risk" individuals in a population. This may comprise only a small portion of the overall network. What's more, this subpopulation may not have access to healthcare, or may be reluctant to get tested, so this portion of the network may be unobservable. Yet we can survey the observable network to identify contributing factors for infection, such as an demographics, genetics and medical history, which may be exhibited in the query set. Moreover, since there is an undeniable causal link between infection and one's proximity to and interaction with those infected, identifying the infection status of related or connected individuals may offer insight about the query set.

Viral Marketing Suppose we are introducing a new product and are interested in creating awareness of it through viral marketing. Given the recent proliferation of online social networks, there are various means of identifying key opinion leaders and information hubs (i.e., the query set), who comprise the optimal entry points into a market. Yet before advertising to them, we must predict whether these individuals are likely to adopt and promote our product. Receiving positive reviews would be beneficial, but having opinion leaders disseminate negative feedback would be especially detrimental to sales. As before, we can survey a less influential test market to model the behavior of the target market without risk of negative publicity. We can also look at how people that are connected to the opinion leaders react to the product, with the assumption that they likely share similar opinions. Using their estimated reactions to the product, we can target our marketing to the subset of opinion leaders likely to give positive reviews, while minimizing the overall marketing cost.

3. Background

For the following, let $\mathcal{X} \subseteq \mathbb{R}^d$ denote a d-dimensional instance space, $\mathcal{Y}$ a finite set of labels, and $\mathcal{Z} \triangleq \mathcal{X} \times \mathcal{Y}$ their cross-product. We are given a relational graph $G = (\mathcal{V}, \mathcal{E})$, in which the nodes $\mathcal{V}$ represent individuals and the edges $\mathcal{E}$ represent relationships between them. We assume that $\mathcal{V}$ is fully-specified, although $\mathcal{E}$ is presumably incomplete. Each node is associated with a vector of attributes $v.X \in \mathcal{X}$ and a label $v.Y \in \mathcal{Y}$, which initially unknown.

We define a *relational* learning algorithm $\mathcal{A}$ as a function mapping an input graph G to a hypothesis space $\mathcal{F}$. Let f_G denote a hypothesis returned by running $\mathcal{A}$ on G, and note that f_G can leverage any information revealed during training to perform collective inference. Accordingly, we denote the prediction of a single instance $v \in \mathcal{V}$ by $f_G(v)$. If f_G is *real-valued* (confidence-rated or probabilistic), we will use $f_G(v; y)$ to denote the predicted confidence (or probability) that $v.Y = y$. (If f_G outputs a probability distribution, then we require that $\sum_{y \in \mathcal{Y}} f_G(v; y) = 1$.) Accordingly, we use $h_G(v)$ to denote the *maximum a posteriori* (MAP) assignment $h_G(v) \triangleq \text{argmax}_{y \in \mathcal{Y}} f_G(v; y)$.

We measure the error (or *loss*) of f_G by a function $\ell : \mathcal{F} \times \mathcal{V} \to \mathbb{R}$, which returns a real-valued measure of the discrepancy between $f_G(v)$ and $v.Y$. Denote by $\mathcal{L}(\mathcal{U}) \triangleq \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \ell(f_G(u))$ the average loss over a subset of nodes $\mathcal{U} \subseteq \mathcal{V}$. This can be equivalently stated as $\mathbb{E}_{u \in \mathcal{U}}[\ell(f_G, u)]$.

3.1. Collective Classification

The task of inferring node labels of network data using local and global structural information is generally known as *collective classification*. The underlying assumption of collective classification models is that the relationships between nodes can be used to supplement local information (attributes) used in prediction. For instance, a node's label might be positively or negatively correlated with that of its neighbors. Some collective methods rely solely on this structural information to propagate labels (Macskassy & Provost, 2007). A number of collective classification models have been proposed (Sen et al., 2008) and shown to outperform their non-relational counterparts in relational domains. This is especially true in semi-supervised settings like ours, in which labeled and unlabeled instances are connected in the same network (Bilgic et al., 2010).

3.2. Active Learning and Inference

While most prior work in collective classification has focused on the "passive" setting, in which labeled data is drawn randomly from an unknown distribution, we consider the "active" setting, in which the learning algorithm (or predictor) can determine the sequence of examples. The learner is given an initial set of annotations with which to bootstrap learning (or inference), after which it is allowed to request additional examples (subject to some budget constraint) to improve performance. In active learning, the benefit is two-fold: by selecting the most informative examples, the learner can refine the model for problematic or ambiguous instances, while potentially reducing the sample complexity of the learning algorithm (Bilgic et al., 2010; Macskassy, 2009). In the transductive setting, where the labeled and unlabeled instances belong to the same network, additional labeled instances can inform the predictions of related nodes, in a process commonly referred to as *active inference* (Bilgic & Getoor, 2010; Rattigan et al., 2007).

Prior work in active learning (Bilgic et al., 2010; Kuwadekar & Neville, 2011; Macskassy, 2009) for relational data has focused on acquiring only label information, with the assumption that the network and all other attributes are observed. Here, we make no such assumptions; instead, we explicitly assume that the available network is largely incomplete. We therefore allow the learner (or classifier) to obtain a richer form of feedback, including (but not limited to) labels, attributes, and network structure. In the context considered herein, we begin with a partially labeled network, with partially specified neighborhoods; surveying any node returns its label, along with any edges connected to it. There may also be contexts in which a survey returns the ground truth for missing or noisy attribute values. Because this form of data acquisition is more general than traditional active learning, we refer to it as "active surveying" (Sharara et al., 2011).

3.3. Active Strategies

The effectiveness of active methods is largely predicated on the strategy for acquiring new information. The goal is to select a sequence of surveys that maximizes the quality of the learned model, while minimizing the amount, or cost, of the acquired information. Since determining an optimal solution is often intractable (Bilgic & Getoor, 2010; Roy & McCallum, 2001), active methods typically rely on heuristics. Popular strategies for active learning and inference are *uncertainty sampling* and *structure-based sampling*, respectively.

Reasoning that instance ambiguity leads to error, uncertainty sampling focuses attention on those instances that the current model finds most difficult to classify. In classification, this requires either confidence-rated prediction or an ensemble of classifiers. There are numerous measures of uncertainty (e.g., *entropy*). Since deterministically selecting the most uncertain instances can sometimes result in exploring outlier regions of the instance space (Saar-Tsechansky & Provost, 2004), uncertainty-based methods typically perform random sampling, weighted by uncertainty, which increases robustness to outliers.

Another broad category of strategies leverages the structure of the network (Bilgic et al., 2010; Rattigan et al., 2007). These approaches rely on the assumption that, during inference, the true labels of nodes with certain structural properties are likely to propagate and positively impact the inference of the most nodes. One heuristic, for example, is to survey the nodes with highest degree, with the intuition that these nodes have the greatest influence over the connected nodes. In other words, the labels of high degree nodes are likely to correlate with those of their neighbors (Rattigan et al., 2007). Other common heuristics include various centrality measures such as closeness and betweenness centrality (Macskassy, 2009) with the assumption that nodes most central to a given connected component are most likely to provide the most influence over nodes in that connected component.

4. Query-driven Active Surveying

In this section, we define the problem of query-driven collective classification with active surveying. We motivate the discussion of surveying strategies by introducing the notion of smoothness. We then leverage the smoothness assumption to derive several active surveying strategies.

4.1. Problem Definition

The learning problem is defined as follows. In query-driven applications, we are given a specified (proper) subset of the full vertex set, $\mathcal{Q} \subset \mathcal{V}$. We refer to this set as the *query set*. Let $\mathbb{Q}$ denote the distribution over this subset and note that it is assumed to be different from the global distribution $\mathbb{P}$. The labels of the query set are hidden and assumed to be unobtainable; thus *our primary objective is to predict the labels of this subset.* To do so, we will train a transductive model, leveraging the label and structural information from the rest of the network.

We obtain training data via a sequence of *surveys*. Each survey returns the label of, as well as all edges adjacent to, a specified node. Let Ψ denote the survey operator. Thus, surveying a node completely reveals all information about the node; until a node is surveyed, one cannot assume that its adjacent edge set is completely specified. Let $\mathcal{S}$ denote the set of nodes that have been surveyed and $\mathcal{U}$ denote the nodes that have yet to be surveyed. When considering which nodes to survey, we may refer to a subset $\mathcal{U}^c \subseteq \mathcal{U}$ as the *survey candidates*.

Acquiring complete information is considered expensive; we therefore assume some cost structure associated with surveying. Let $\varphi : \mathcal{V} \rightarrow \mathbb{R}^+$ denote a real-valued cost function. For the nodes in the query set, the cost is infinite[1]; for all other nodes, the cost is a positive real number. For the purposes of this research, since our study focuses on the efficacy of our survey strategies, we will assume that the cost of a survey is uniform for all non-query nodes.

Our learning objective can be stated as the cost of the queries and the expected loss over the query set:

$$\operatorname{argmin}_{\mathcal{S}} \mathbb{E}_{q\in\mathcal{Q}}\left[\ell(f_G,q)\,|\,G\leftarrow G\cup\mathcal{S}\right]+\sum_{s\in\mathcal{S}}\varphi(s).$$

Determining the optimal set of surveys is obviously hard, since we cannot measure the expected error term. Even if we could measure the objective, the problem is equivalent to exactly solving a *knapsack problem*, which is NP-hard. As such, we consider an iterative greedy approach, in which we survey a fixed number of nodes at each time step. Without loss of generality, assume for the moment that we survey one node at a time; at each iteration, the objective is

$$\operatorname{argmin}_{u\in\mathcal{U}} \mathbb{E}_{q\in\mathcal{Q}}\left[\ell(f_G,q)\,|\,G\leftarrow G\cup\Psi(u)\right]+\varphi(u).$$

Still, we cannot measure this objective. We discuss heuristics to approximate it in the following section, and address surveying strategies based on these heuristics in subsection 4.3.

4.2. The Smoothness Assumption

To motivate the discussion of survey strategies, we examine the following scenario. Recall that $\mathcal{Q}$ is the set of query nodes and $\mathcal{S}$ the surveyed nodes, and let $\mathbb{Q}$ and $\mathbb{S}$ denote their respective empirical distributions. That is, for a random variable Z taking values in $\mathcal{Z}$, $\mathbb{Q}(Z)=\Pr[Z\in\mathcal{Q}]$, and similarly for $\mathbb{S}$. If the loss is bounded by M for any $z\in\mathcal{Z}$, then by the triangle inequality, we have that

$$\begin{aligned}
&\mathbb{E}_{q\in\mathcal{Q}}\left[\ell(f_G,q)\,|\,G\right]\\
&\qquad=\sum_{z\in\mathcal{Z}}\ell(f_G,z)\left(\mathbb{Q}(z\,|\,G)-\mathbb{S}(z\,|\,G)+\mathbb{S}(z\,|\,G)\right)\\
&\qquad\leq \mathbb{E}_{s\in\mathcal{S}}\left[\ell(f_G,s)\,|\,G\right]+M\sum_{z\in\mathcal{Z}}\left|\,\mathbb{Q}(z\,|\,G)-\mathbb{S}(z\,|\,G)\,\right|\\
&\qquad=\mathbb{E}_{s\in\mathbb{S}}\left[\ell(f_G,s)\,|\,G\right]+M\left|\left|\,\mathbb{Q}(Z\,|\,G)-\mathbb{S}(Z\,|\,G)\,\right|\right|_{\mathrm{TV}},
\end{aligned}\tag{1}$$

where $||\cdot||_{\mathrm{TV}}$ is the *total variation norm*. We interpret Equation 1 to mean that the difference between the average errors over $\mathcal{Q}$ and $\mathcal{S}$ is a function of the *statistical distance* between their respective distributions. Furthermore, note that $\mathbb{E}_{s\in\mathcal{S}}\left[\ell(f_G,s)\,|\,G\right]$ is an empirically measurable quantity, which is (typically) minimized by the learning algorithm. Thus, in order to minimize the error over $\mathcal{Q}$, we must not only minimize the empirical error over $\mathcal{S}$, but also survey nodes such that the $\mathbb{S}$ becomes "close" to $\mathbb{Q}$.

Since the labels of $\mathcal{Q}$ and the unsurveyed set $\mathcal{U}$ are hidden, deciding which subset $\mathcal{S}$ will minimize the distance between $\mathbb{Q}$ and $\mathbb{S}$ is hard. Fortunately, intuition offers a solution in the form of *distributional smoothness*. A common assumption in semi-supervised learning is that the distribution over the instance space is "smooth"—that is, high density areas are likely to exhibit the same labels. This assumption has been used to explain the effectiveness of instance-based methods, such as k-nearest neighbors (Cover & Hart, 1967) and various semi-supervised approaches (Zhu & Goldberg, 2009). We can adapt this reasoning to the query-driven setting. Let P be some property associated with each node, taking values in a space $\mathcal{P}$. For instance, a specific feature value, or perhaps its encoded location in the network. We say that a query-driven problem is *smooth with respect to a distance function* d if there exists a constant $\beta\geq 0$ such that, for any $p,p'\in\mathcal{P}$,

$$\left|\left|\Pr_{v\in\mathcal{V}}[v\,|\,v.P=p]-\Pr_{v\in\mathcal{V}}[v\,|\,v.P=p']\right|\right|_{\mathrm{TV}}\leq\beta\,d(p,p').\tag{2}$$

In other words, the statistical distance[2] between the conditional distributions of a node with property p versus a node with property p' should be bounded by a constant multiplier of the distance between p and p'. Equation 2 suggests a strategy for minimizing the distance between $\mathbb{Q}$ and $\mathbb{S}$ without having access to the labels: *if the smoothness property holds for a given distance function, then survey nodes in $\mathcal{U}$ that have minimal distance to nodes in $\mathcal{Q}$.*

Identifying a distance function for which the smoothness assumption holds is a fundamental challenge in the query-driven setting. There are a number of metrics to choose from, and the appropriateness of any given one depends on the data. We emphasize the fact that smoothness is an *assumption* that we make about a particular problem. Indeed, in certain applications, this assumption may not hold for *any* metric. Yet it is reasonable to assume that it does hold in certain cases, given insight into the problem domain.

Feature Smoothness A common assumption in data analysis is that the distribution exhibits smoothness with respect to a similarity or distance function in feature space. In the query-driven setting, we can assume that nodes that are similar (or close) in feature space will exhibit similar label distributions; in other words, the problem is smooth with respect to attribute similarity (or distance).

The exact nature of the similarity or distance function is context-specific. One popular similarity measure for arbi-

[1] While in certain settings query nodes may trivially be surveyed directly, we focus on the more challenging setting where nodes in the query set cannot be surveyed.

[2] One could define smoothness using an alternate notion of statistical distance. In this case, the total variation norm fit nicely with the preceding analysis.

trary vectors is *cosine similarity*, with Euclidean distance as the associated distance function. This has been shown particularly effective with text data represented as TF/IDF-weighted word frequencies (Manning et al., 2008).

Structural Smoothness A common assumption in relational domains is that the labels of related (i.e., connected) nodes are correlated. Collective methods have been shown to outperform traditional local models because they can exploit these correlations (e.g., (Sen et al., 2008)). Consequently, a natural similarity criterion for network data is adjacency.

Since the structure of the network may be only partially observed, there may be few direct adjacencies to the query set. One can address this problem by also applying a *link predictor* to the graph. Much work has been done on this topic, resulting in learning algorithms to infer the existence of missing edges. In practice, we found these techniques to be too computationally expensive to apply in the iterative active setting. If these methods are too expensive, one can use a simpler, path-based link predictor instead. One such method (Liben-Nowell & Kleinberg, 2003) is the *Katz score*. Note that this is a purely structural measure, whose effectiveness cannot be explained by attribute similarity. Furthermore, since it will tend to assign higher scores to directly adjacent nodes, it provides an easy way to integrate observed edges; one can therefore use the Katz score as a single indicator of both observed and inferred adjacency.

4.3. Survey Strategies

Given a budget of k surveys, we use the idea of smoothness to decide which nodes to survey. Under the smoothness assumption, we expect high utility from nodes that are close (with respect to a metric d) to the query nodes $\mathcal{Q}$. This invokes two questions: (1) how to compute utility for each unsurveyed node; (2) how to sample within the budget.

To address the first question, we could compute an aggregate utility value for each $u \in \mathcal{U}$ by summing $d(q, u)$ over all $q \in \mathcal{Q}$. However, since $\mathcal{Q}$ may exhibit high variance, the aggregated utility may yield little overall benefit. For example, suppose that $\mathcal{Q}$ lies on the surface of a multidimensional sphere (in feature space); applying an aggregate feature similarity will result in selecting nodes at the middle of the sphere, which, while equidistant to all query nodes, may not be as informative as those closer to the perimeter. As such, instead of computing an aggregate utility, we could sample from the full cross-product of $\mathcal{Q} \times \mathcal{U}$ according to which u is the best proxy for each q. For each $q \in \mathcal{Q}$, we compute the utility of every $u \in \mathcal{U}$ with respect to q, then add the highest scoring u to a pool of survey candidates $\mathcal{U}^c$. The usefulness of each survey candidate is thus conditioned on a particular query node, instead of over all query nodes. Interpreted differently, the utility measures the amount of proxy information for a *specific* query node.

Algorithm 1 Adaptive Query-driven Active Surveying for Collective Classification (QD$_{\text{Adapt}}$)

```
Input: Initial network $G = (\mathcal{V}, \mathcal{E})$; set of query nodes $\mathcal{Q}$; cost function $\varphi$; feature similarity $d_{\text{fs}}$; structural similarity $d_{\text{ss}}$; survey budget $B$; survey batch size $k$.
Output: the surveyed network $G$.
$\mathcal{S} \leftarrow \emptyset,\ \mathcal{U} \leftarrow \mathcal{V}$
while $B > 0$ do
  $\alpha \leftarrow$ (Re-)Estimate assortativity of G
  With probability $p = |\alpha|$, $d \leftarrow d_{\text{ss}}$; else $d \leftarrow d_{\text{fs}}$
  $\mathcal{U}^c \leftarrow \emptyset$
  for $q \in \mathcal{Q}$ do
    $u_q \leftarrow \text{argmax}_{u \in \mathcal{U} \setminus (\mathcal{U}^c \cup \mathcal{Q})} d(q, u)$
    Add $u_q$ to $\mathcal{U}^c$ with weight $d(q, u_q)$
  end for
  $\mathcal{U}^s \leftarrow$ Weighted sampling of $k$ nodes from $\mathcal{U}^c$
  for $u \in \mathcal{U}^s$ do
    $G \leftarrow G \cup \Psi(u)$
    $\mathcal{S} \leftarrow \mathcal{S} \cup u,\ \mathcal{U} \leftarrow \mathcal{U} \setminus u$
    $B \leftarrow B - \varphi(u)$
  end for
  $f_G \leftarrow \mathcal{A}(G)$
end while
```

Given $\mathcal{U}^c$ and a budget constraint of k surveys, we must determine how to sample from this set. Assuming the utility function is perfect, we could just select the top-k nodes. Yet since the utility is predicated on an *assumption* about the data, a deterministic selection might yield suboptimal results. For this reason, we propose introducing stochasticity by performing a weighted random sampling according to utility.

To summarize, for each query node, we select its proxy from the pool of unsurveyed nodes, based on the given utility (i.e., distance) function, and flag it as a survey candidate. From the pool of survey candidates, we then perform a weighted sampling, proportional to the utility. The following section introduces an adaptive surveying strategy to combine feature- and structure-based criteria.

4.4. An Adaptive Survey Strategy

Any smoothness assumption—be it feature-based, structural, or otherwise—is only an assumption, and is wholly data-dependent. There is no single distance function that will always work. That said, given a set of potentially useful metrics, one can adaptively select the best one for the given problem and current information.

We develop an active surveying algorithm to adaptively choose between feature-based and structural metrics. This algorithm uses a novel mechanism for determining when to trust structural measures by using the *assortativity* (Newman, 2003) of the currently observed graph. Let e_y be the fraction of edges in the network that connect two nodes of class y. Let s_y be the fraction of edges with source nodes that are in class y. Similarly, let t_y be the fraction of destination nodes in class y. The assortativity of a graph is defined as:

$$\text{assortativity}(G) = \frac{\sum_{y \in \mathcal{Y}} e_y - \sum_{y \in \mathcal{Y}} s_y t_y}{1 - \sum_{y \in \mathcal{Y}} s_y t_y}.$$

Informally, assortativity is a measure of how correlated the nodes in a network are. We use this as an indicator of when there is sufficient correlation to use the structural similarity as the distance function. More specifically, with probability equal to the absolute value[3] of the assortativity, we decide to exploit the structural smoothness; otherwise, we use the feature smoothness. Note that because the labels of most nodes and edges are initially unobserved, we cannot compute assortativity of the fully observed graph exactly. We instead estimate the assortativity of the currently observed graph using the observed edges and both the observed and predicted labels. The rest of the algorithm follows the strategy outlined in subsection 4.3. The details of the QD$_{\text{Adapt}}$ algorithm are shown in Algorithm 1.

5. Empirical Evaluation

We evaluate our approach using several benchmark collective classification datasets. We begin by describing the characteristics of these networks, and our general experimental setup. We evaluate our active surveying strategies on these networks and compare the performance to active learning approaches.

5.1. Experimental Setup

In these experiments, we use four real-world networks: CORA, CITESEER, WIKIPEDIA, and PUBMED[4]. The first two, CORA and CITESEER, are benchmark collective classification networks of computer science publications. In these publication networks, each node represents a publication and each edge a citation. Each node is annotated with a vector of binary word indicators (i.e., whether it contains each word) and a label indicating the paper topic. The WIKIPEDIA network consists of Wikipedia articles, wherein each node represents an article and each edge a hyperlink between articles. Each node is annotated by a vector of TF/IDF-weighted word frequencies and a label specifying the general category. Finally, the PUBMED citation network is a set of articles related to diabetes from the PubMed database. Node attributes are TF/IDF-weighted word frequencies and the labels specify the type of diabetes addressed in the publication.

For each dataset, we limit our experiments to the largest connected component. For the purposes of collective classification, we ignore the directionality of hyperlinks and citations. To prepare the word attribute data, we use stemming, stop-word removal, and filter for the highest TF/IDF-weighted words to reduce the size of the dictionary to 500. In all of our experiments, the learning algorithm receives a partially observed network where the node labels are hidden, but the node features, a random 10% of the edges, and attributes are observed. Whenever a node is surveyed, the learner acquires the node's label and its incident edges.

5.2. Methodology

We compare our adaptive query-driven approach (QD$_{\text{Adapt}}$) to two commonly used active learning baseline strategies: uniform random sampling (RAND) from the unsurveyed nodes $\mathcal{U}$, and weighted uncertainty sampling (UNC) over the $\mathcal{U}$ based on entropy (Saar-Tsechansky & Provost, 2004). We also compare to variants of QD$_{\text{Adapt}}$ which only exploit one of the smoothness types each: QD$_{\text{FS}}$ for feature smoothness and QD$_{\text{SS}}$ for structural smoothness. As mentioned in Section 4.2, we use cosine similarity for feature smoothness and the approximate *Katz score* for structural smoothness. We perform active surveying in *batch-mode* (Settles, 2009), common in active learning problems, where k nodes are surveyed in parallel. We set the survey batch size $k = 10$ and allow the algorithm to run for 30 iterations (yielding an effective budget of 300 surveys).

Our algorithm is largely agnostic to the underlying collective classification model. For our experiments, we use a semi-supervised variant of the Iterative Classification Algorithm (ICA) (Bilgic et al., 2010) to perform the collective classification. In ICA, each node is annotated with a vector of its attribute values (i.e., words), its label, and the label distribution of its neighbors. ICA learns two base classifiers: a local classifier and a relational classifier. The local classifier, trained on the observed labels using only the attribute values, is used to bootstrap the unobserved labels prior to learning the relational classifier. The local classifier is also used to bootstrap the unobserved labels prior to applying the relational classifier during inference. The relational classifier, trained on the observed labels using the attribute values and neighbor label distribution, is then iteratively applied during inference to propagate the labels. Any classifier (e.g., logistic regression, naïve bayes, support vector machines) can be used for the base classifiers. We use support vector machines with a linear kernel (Chang & Lin, 2001) for both base classifiers.

To evaluate our approaches under different conditions, we explore various query set generating processes. We evalu-

[3] The assortativity ranges from -1 to 1: positive scores indicate correlation, and negative scores indicate anticorrelation. In either case, the magnitude is the quantity we are interested in, as it indicates the level of structural smoothness.

[4] Datasets available from: `http://www.cs.umd.edu/projects/linqs/projects/lbc`.

ate both on query sets that are generated by uniform random sampling and query sets generated by targeting a particular structural or attribute characteristics, described in greater detail below.

5.3. Sampled Query Sets

For our first set of experiments, we create query sets by randomly sampling (uniformly and without replacement) 5% of the nodes. Table 1 lists the number of iterations that QD$_{\text{Adapt}}$ outperforms each other method on average, and lists in parentheses the number of times the improvement by QD$_{\text{Adapt}}$ is statistically significant via a paired t-test. First, we find that for all networks, our query-driven approaches typically outperform RAND and UNC, the two non-query-driven baselines. QD$_{\text{Adapt}}$ performs best for over a majority of the budgets considered, with most of these gains deemed statistically significant. Specifically, QD$_{\text{Adapt}}$ achieves performance improvements of up to 17% over RAND and UNC. It is important to note that neither QD$_{\text{FS}}$ nor QD$_{\text{SS}}$ performs uniformly well on *all* datasets, thus motivating the adaptive strategy of QD$_{\text{Adapt}}$. We also find that the structural distance criterion works well for Cora, Citeseer, and PubMed; this is likely due to the fact that paper topic is typically correlated across citations. In these datasets, attribute similarity is not as strong an indicator, and so QD$_{\text{FS}}$ does not perform as well. However, in the Wikipedia dataset, we find that QD$_{\text{FS}}$ performs very well, while QD$_{\text{SS}}$ performs the worst; this is likely due to the fact that Wikipedia articles often link to a large number of unrelated articles, whereas their word frequencies are better indicators of topic. Analyzing the true assortativity of these datasets supports this claim. We find that Cora, Citeseer, and PubMed have high assortativities with respective values of 0.79, 0.67 and 0.69; meanwhile, Wikipedia has a low assortativity of 0.36.

Focusing on the query-driven strategies, we find that QD$_{\text{Adapt}}$ generally outperforms both QD$_{\text{FS}}$ and QD$_{\text{SS}}$ on all citation networks by as much as 12% and 8% respectively. Only on the Wikipedia dataset did a non-adaptive strategy generally outperform our adaptive approach, typically in the early iterations (i.e., low survey budgets); and even in this case, QD$_{\text{Adapt}}$ is still competitive. We note, however, that the non-adaptive strategies are only useful if we know *a priori* which metric to use in advance, which is rarely the case in practice.

Table 1. Number of iterations (out of 30) where QD$_{\text{Adapt}}$ scores higher on average (wins) or lower (losses) than each other method. Of those, the number of significant wins and losses, using paired t-tests with 90% significance, are listed in parentheses.

		# of Wins		# of Losses	
Cora	RAND	29	(28)	1	(0)
	UNC	29	(27)	1	(0)
	QD$_{\text{FS}}$	30	(25)	0	(0)
	QD$_{\text{SS}}$	24	(21)	6	(0)
Citeseer	RAND	30	(20)	0	(0)
	UNC	30	(23)	0	(0)
	QD$_{\text{FS}}$	30	(18)	0	(0)
	QD$_{\text{SS}}$	24	(23)	6	(2)
Wikipedia	RAND	24	(9)	6	(1)
	UNC	26	(11)	4	(1)
	QD$_{\text{FS}}$	4	(0)	26	(7)
	QD$_{\text{SS}}$	28	(26)	2	(0)
PubMed	RAND	27	(17)	3	(0)
	UNC	26	(18)	4	(1)
	QD$_{\text{FS}}$	26	(14)	4	(1)
	QD$_{\text{SS}}$	22	(1)	8	(0)

5.4. Targeted Query Sets

In practice, query sets are selected for some context-specific reason, and thus may have certain targeted characteristics. For example, in the disease transmission example of Section 2, where physical contact is a significant factor, query nodes may tend to be highly interconnected. Similarly, in the viral marketing example of Section 2, query nodes may share a common characteristic such as being popular or prolific. To study the impact of more targeted generating processes, we next generate query sets with two types of targeted queries: neighbor-based and characteristic-based queries.

To generate neighbor-based queries, we select nodes using *snowball sampling*. In snowball sampling, we initialize the query set using a *seed* node. We then proceed to sample each of its neighbors with probability p_{neigh}; if we do not sample a neighbor (which occurs with probability $1 - p_{\text{neigh}}$), then we select a random node from the remaining unsampled network. We repeat this process for each node currently in the query set, until the number of query nodes reaches 5% of the overall network. We perform this procedure for $p_{\text{neigh}} = 0.1, 0.5, 0.9$. Note that, for higher values of p_{neigh}, the query set tends to be a connected component. Conversely, for lower values of p_{neigh}, the query set tends to be randomly distributed throughout the network. We test this neighbor-based setup using the Citeseer network repeating the experiment for 40 runs by sampling query sets using different random seeds.

To recreate a query sets based on common characteristic, we first identify a set of words such that the probability of occurrence is low (below 5%) and which a domain expert may find interesting. We then generate the query set from all documents that contain the word. For this set of experiments, we focus on the PubMed network. We used domain knowledge to select words such as “death”, “hypoglycemia”, and “suppress” as the criteria for adding a paper to the query set.

Examining the results, we see similar trends as before, with

$\mathrm{QD}_{\mathrm{Adapt}}$ showing even greater improvement over the baselines. Two important observations when comparing the targeted query-sets setting with the random query-set setting. First, while $\mathrm{QD}_{\mathrm{Adapt}}$ is still overall the best performing, there are cases where either $\mathrm{QD}_{\mathrm{FS}}$ or $\mathrm{QD}_{\mathrm{SS}}$ outperform $\mathrm{QD}_{\mathrm{Adapt}}$ on targeted query sets. We see this change when comparing low and high values of p_{neigh} and when comparing the results between the randomly generated and attribute-based query sets. The effectiveness of the non-adaptive smoothness heuristics is especially noticeable when the number of surveyed nodes is particularly small (i.e., the learner's budget is small). This effect implies that when budget is particularly low for query sets that exhibit clear biases, and there is domain knowledge that can identify in advance whether feature or structure smoothness is more likely, using either $\mathrm{QD}_{\mathrm{FS}}$ or $\mathrm{QD}_{\mathrm{SS}}$ alone can potentially yield better results. For most greater budgets, however, and in the absence of prior knowledge about the general characteristics of the data, $\mathrm{QD}_{\mathrm{Adapt}}$ generally yields the best performance.

Next, we observe a general upward trend when comparing the results from the randomly generated query sets to targeted query sets. In both cases, the stronger the bias for the query set sampling, the greater the improvement over the non-query-driven strategies. For example, while the percent improvements of $\mathrm{QD}_{\mathrm{Adapt}}$ over RAND and UNC reach up to 12% and 17% for uniformly random query sets, we find improvement as great as 22% and 68% accuracy for high values of p_{neigh}. Similarly, in the PUBMED experiments, where we reach up to 10% and 11% improvement over RAND and UNC on a uniformly random query set, using $\mathrm{QD}_{\mathrm{Adapt}}$ on query sets defined by the word attributes improves accuracy by up to 28% and 44%. Consequently, while $\mathrm{QD}_{\mathrm{Adapt}}$ already yields significant improvements in the uniformly random query-set setting from the previous section, the results from tests in this section indicate that the more realistic setting where the query nodes are selected based on some measurable criteria will benefit even more.

6. Conclusion

Query-driven collective classification is an important but understudied problem, applicable to a variety of domains. The query-driven setting, when coupled with active surveying for partially observed networks, is natural in practice. It provides an opportunity to develop high impact algorithms for maximizing predictive performance, over a range of annotation budgets. We identify two forms of data smoothness, feature-based and structure-based, and demonstrate how to exploit them for query-driven active surveying. We then develop an adaptive algorithm to automatically determine the optimal smoothness assumption, given the observed information. We evaluate these survey strategies on real network data and show that our query-driven methods exhibit significant advantages over traditional (non-query-driven) active learning heuristics. There is much room for further exploration: for example, query-driven active surveying in which surveys may return incomplete or noisy information; exploring non-uniform cost structures; and application in dynamic networks. Nevertheless, this paper identifies this important and challenging problem setting, and represents a major first step in addressing it.

7. Acknowledgments

This work was supported by DARPA under contract #SC1007202, ARO under grant #W911NF0810466, and NSF grant #IIS0746930.

References

Bilgic, Mustafa and Getoor, Lise. Active inference for collective classification. In *AAAI*, 2010.

Bilgic, Mustafa, Mihalkova, Lilyana, and Getoor, Lise. Active learning for networked data. In *ICML*, 2010.

Chang, Chih-Chung and Lin, Chih-Jen. *LIBSVM: a library for support vector machines*, 2001.

Cover, T. and Hart, P. Nearest neighbor pattern classification. *IEEE Trans. on Information Theory*, 13(1):21 – 27, 1967.

Kuwadekar, Ankit and Neville, Jennifer. Relational active learning for joint collective classification models. In *ICML*, 2011.

Liben-Nowell, David and Kleinberg, Jon. The link prediction problem for social networks. In *CIKM*, 2003.

Macskassy, Sofus A. Using graph-based metrics with empirical risk minimization to speed up active learning on networked data. In *KDD*, 2009.

Macskassy, Sofus A. and Provost, Foster. Classification in networked data: A toolkit and a univariate case study. *JMLR*, 8: 935–983, 2007.

Manning, Christopher D., Raghavan, Prabhakar, and Schtze, Hinrich. *An Introduction to Information Retrieval*. Cambridge University Press, 2008.

Newman, Mark E. J. Mixing patterns in networks. *Physics Review E*, 67(2):026126, Feb 2003.

Pfeiffer III, Joseph J., Neville, Jennifer, and Bennett, Paul N. Active sampling of networks. In *10th International Workshop on Mining and Learning with Graphs*, 2012.

Rattigan, Matthew J., Maier, Marc, Wu, David Jensen Bin, Pei, Xin, Tan, JianBin, and Wang, Yi. Exploiting network structure for active inference in collective classification. In *ICDM Workshops*, 2007.

Roy, Nicholas and McCallum, Andrew. Toward optimal active learning through sampling estimation of error reduction. In *ICML*, 2001.

Saar-Tsechansky, Maytal and Provost, Foster. Active sampling for class probability estimation and ranking. *Machine Learning*, 54(2):153–178, 2004.

Sen, Prithviraj, Namata, Galileo Mark, Bilgic, Mustafa, Getoor, Lise, Gallagher, Brian, and Eliassi-Rad, Tina. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

Settles, Burr. Active learning literature survey. 1648, University of Wisconsin-Madison, 2009.

Sharara, Hossam, Getoor, Lise, and Norton, Myra. Active surveying: A probabilistic approach for identifying key opinion leaders. In *IJCAI*, 2011.

Zhu, Xiaojin and Goldberg, Andrew B. *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers, 2009.