Active Learning for Networked Data Based on Non-progressive Diffusion Model

Zhilin Yang
Department of Computer Science and Technology
Tsinghua University
kimiyoung@yeah.net

Jie Tang and Bin Xu
Department of Computer Science and Technology
Tsinghua University
{jietang, xubin}@tsinghua.edu.cn

Chunxiao Xing
Department of Computer Science and Technology
Tsinghua University
xingcx@tsinghua.edu.cn

ABSTRACT

We study the problem of active learning for networked data, where samples are connected with links and their labels are correlated with each other. We particularly focus on the setting of using the probabilistic graphical model to model the networked data, due to its effectiveness in capturing the dependency between labels of linked samples.

We propose a novel idea of connecting the graphical model to the information diffusion process, and precisely define the active learning problem based on the non-progressive diffusion model. We show the NP-hardness of the problem and propose a method called MaxCo to solve it. We derive the lower bound for the optimal solution for the active learning setting, and develop an iterative greedy algorithm with provable approximation guarantees. We also theoretically prove the convergence and correctness of MaxCo.

We evaluate MaxCo on four different genres of datasets: Coauthor, Slashdot, Mobile, and Enron. Our experiments show a consistent improvement over other competing approaches.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Experimentation

Keywords

Active learning, Non-progressive model, Factor graph model

1. INTRODUCTION

One challenge for a machine learning task is how to collect sufficient labeled samples for training an accurate classification model. Active learning is a method to alleviate this problem by actively querying experts to obtain the desired labels of a few samples. For example, Hoi et al. [12] studied active learning on text categorization problem. Cohn et al. [5] proposed two methods: mixtures of Gaussians and locally weighted regression for efficiently selecting samples in neural networks. Settles et al. [22] provided a survey for various sample selection strategies. The underlying idea for most of these methods is to measure the informativeness of each unlabeled sample and finally select an "informative" sample to query each time. The problem becomes more difficult with the increase of the complexity of the input data. First, the samples in the input data may be connected and correlated with each other (i.e., networked data), which implies that selecting the most informative (but isolated sample) may be not that helpful for classifying the other samples. Second, in practice, to avoid frequently querying the experts, it is usually desirable to select a set of samples and query the users in a batch mode.

In this paper, we try to systematically address the above questions. The problem can be formally defined as follows:

Definition 1. **Batch Mode Active Learning for Networked Data.** Given a network $G = (V_U, V_L, \mathbf{y}_L, E, \mathbf{X})$, where V_U denotes a set of unlabeled samples, V_L denotes a set of labeled samples, $\mathbf{y}_L$ corresponds to labels of the labeled samples, E is the set of edges between samples in the network G, and $\mathbf{X}$ is an $(|V_U| + |V_L|) \times d$ attribute matrix in which each row $\mathbf{x}_i$ represents the vector of attributes for sample v_i, our goal is to query a subset of k unlabeled samples so as to maximize the following utility function:

$$\max_{V_S \subseteq V_U} Q(V_S), \text{ with } \quad |V_S| \leq k \tag{1}$$

In the formulation, the utility function $Q(V_S)$ is a general definition on the subset V_S, and can be instantiated in different ways. Such a definition of active learning for networked data has been extensively used in the literature [2, 23, 31].

To model the correlation between labels of linked samples, we consider the probabilistic graphical model. In the setting of graphical model, we connect the active learning problem to the theory of non-progressive diffusion [8], and develop an instantiation model for the above problem. The active learning problem based on non-progressive diffusion is proved to be NP-hard. We present an efficient method named MaxCo to solve the problem with provable approximation guarantee. Theoretically, we prove the convergence and correctness of the proposed method and also provide its approximation ratio.

Empirically, we verify the proposed method on four different genres of datasets: Coauthor, Slashdot, Mobile, and

WSDM'14, February 24–28, 2014, New York, New York, USA.

http://dx.doi.org/10.1145/2556195.2556223.

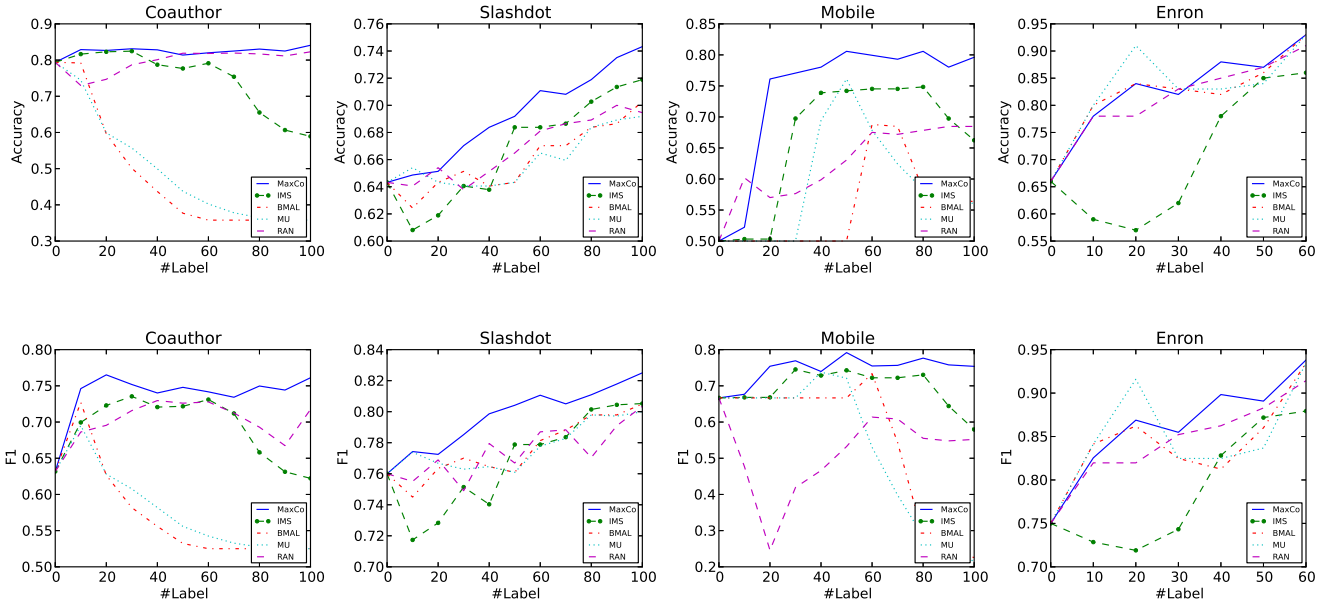


Figure 1: Performance Comparison: Five algorithms on four datasets with accuracy and F1 measure. X-axis denotes the number of labeled samples while y-axis represents the measure score.

Enron. We compare MaxCo with several competing methods (Cf. §5 for definitions of the competing methods). The experimental results indicate that the proposed method can significantly improve the active learning performance over the other methods. Figure 1 shows the performance comparison of the comparison methods on the four datasets. On average, our method MaxCo outperforms the other methods by 5%-10% in terms of F1 (t-test, p-value < 0.01).

To summarize, the major contribution of this work lies in the following aspects:

- Precisely define the problem of active learning for networked data via the non-progressive diffusion model, which provides an elegant way to model the diffusion probabilities from the available labeled samples to the unlabeled samples.
- Prove the NP-hardness of the problem and develop an efficient algorithm with provable approximation guarantee to solve the problem.
- Theoretically analyze the proposed algorithm, prove its convergence and correctness, and provide an upper bound and a lower bound of the proposed algorithm.
- Empirically evaluate the proposed algorithm on several real-world datasets to demonstrate the effectiveness of our algorithm.

Organization. Section 2 presents the factor graph framework and instantiates the problem in the settings of non-progressive diffusion model; Section 3 proposes the algorithm; Section 4 provides the theoretical analysis; Section 5 evaluates the algorithm; Section 6 discusses related work, and finally Section 7 concludes the work.

2. MODEL

The first question we want to address is how to leverage the link information to improve the effectiveness of active learning. In a network setting, for example a social network, users are connected with each other and their behaviors are strongly correlated. To deal with this, we consider the probabilistic graphical model as our basic framework. We utilize Loopy Belief Propagation (LBP) [17] to learn parameters in the probabilistic graphical model.

For active learning, a frequently used method is to select the most informative samples [23, 31]. However, they do not consider the fact that in the graphical model learning process, an instance classified to have label y_i after iteration τ may be classified to have label y_j after the next iteration $\tau + 1$. To this end, we propose a novel idea of connecting message passing in LBP to the theory of non-progressive diffusion and instantiate the active learning problem based on non-progressive diffusion model.

2.1 Factor Graph Model

Factor graph is one type of probabilistic graphical models. It leverages factorization of probabilistic distribution to capture dependency and correlation among random variables.

In particular, we consider a partially labeled setting for the factor graph model, denoted as $G = (V_L, V_U, \mathbf{y}_L, E, \mathbf{X})$, which is consistent with our definition in active learning problem for networked data. We associate each sample $v_i \in V_U \cup V_L$ with a random variable y_i taking value in a discrete space $\mathcal{Y}$. Basically, there are two categories of nodes in a factor graph, variable nodes and factor nodes. If a factor function is defined over a clique, a factor node will be added into the graph and all variable nodes in the clique will be connected to the factor node respectively. Given this, let $V = V_U \cup V_L$, we can define the joint probability over all the labeled and unlabeled samples:

$$P(\mathbf{y}|\mathbf{y}_L;\Theta) = \frac{1}{Z}\exp\left\{\sum_{v_i\in V}\theta_i f(y_i,\mathbf{x}_i) + \sum_{(v_i,v_j)\in E}\theta_{i,j}g(y_i,y_j)\right\} \quad (2)$$

where $f(y_i, \mathbf{x}_i)$ represents the factor function defined on variable node y_i with attribute vector $\mathbf{x}_i$ and $g(y_i, y_j)$ denotes the factor function defined on a factor node that connects y_i and y_j; $\Theta = (\{\theta_i\}, \{\theta_{ij}\})$ are parameters to be estimated and Z is a normalization factor.

Model Learning. Given a training dataset, we aim to estimate the parameter Θ in the factor graph model. One challenge here is how to leverage unlabeled data for parameter estimation. When training the parameter, we maximize the likelihood of labeled samples by summing up all possible distributions over the unlabeled samples. For the sake of simplicity, we first rewrite the conditional probability (Eq. 2) as follows:

$$P(\mathbf{y}|\mathbf{y}_L; \Theta) = \frac{1}{Z} \exp\{\Theta^T \mathbf{S}(\mathbf{y})\} \tag{3}$$

where $\mathbf{S}(\mathbf{y})$ denotes all the factor functions defined on the graph G related to variable $\mathbf{y}$. For maximum likelihood estimation, the log-likelihood objective function can be written as:

$$\begin{aligned} \mathcal{O} &= P(\mathbf{y}_L|G) \\ &= \log \sum_{\mathbf{y}|\mathbf{y}_L} \exp\{\Theta^T \mathbf{S}(\mathbf{y})\} - \log Z \end{aligned} \tag{4}$$

where $\mathbf{y}|\mathbf{y}_L$ indicates a label configuration inferred from the available labels $\mathbf{y}_L$. Then we can write the gradient of the objective function w.r.t. to the parameter Θ:

$$\begin{aligned} \frac{\partial \mathcal{O}}{\partial \Theta} &= \frac{\partial}{\partial \Theta} \log \sum_{\mathbf{y}|\mathbf{y}_L} \exp\{\Theta^T \mathbf{S}(\mathbf{y})\} - \frac{\partial}{\partial \Theta} \log \sum_{\mathbf{y}} \exp\{\Theta^T \mathbf{S}(\mathbf{y})\} \\ &= \frac{\sum_{\mathbf{y}|\mathbf{y}_L} \exp\{\Theta^T \mathbf{S}(\mathbf{y})\} \cdot \mathbf{S}(\mathbf{y})}{\sum_{\mathbf{y}|\mathbf{y}_L} \exp\{\Theta^T \mathbf{S}(\mathbf{y})\}} - \frac{\sum_{\mathbf{y}} \exp\{\Theta^T \mathbf{S}(\mathbf{y})\} \cdot \mathbf{S}(\mathbf{y})}{\sum_{\mathbf{y}} \exp\{\Theta^T \mathbf{S}(\mathbf{y})\}} \\ &= \mathbb{E}_{p(\mathbf{y}|\mathbf{y}_L;\Theta)} \mathbf{S}(\mathbf{y}) - \mathbb{E}_{p(\mathbf{y};\Theta)} \mathbf{S}(\mathbf{y}) \end{aligned} \tag{5}$$

We can apply gradient descent method to solve the objective function. However, $\mathbb{E}_{p(\mathbf{y}|\mathbf{y}_L;\Theta)}$ and $\mathbb{E}_{p(\mathbf{y};\Theta)}$ are intractable when the graph contains cycles [29]. A state-of-the-art approximate solution is Loopy Belief Propagation(LBP)[17].

Loopy Belief Propagation. LBP utilizes message passing to calculate marginal probability. More specifically, for each iteration, message passing is performed according to the following update rules.

$$\begin{aligned} \mu^{\tau}_{y_i \to f}(\mathbf{x}_i) &= \prod_{f^* \in NB(y_i) \setminus f} \mu^{\tau-1}_{f^* \to y_i}(\mathbf{x}_i) \\ \mu^{\tau}_{f \to y_i}(\mathbf{x}_i) &= \sum_{\sim y_i} f(\mathbf{x}_f) \prod_{y_j \in NB(f) \setminus \{y_i\}} \mu^{\tau-1}_{y_j \to f}(\mathbf{x}_j) \end{aligned} \tag{6}$$

For iteration τ, $\mu^{\tau}_{y \to f}$ represents the message passed from a variable node y to a factor node f, while $\mu^{\tau}_{f \to y}$ denotes the message passed in the reverse direction. $NB(y)$ is the set of neighbor factor nodes of a variable node y. $f(\mathbf{x}_f)$ denotes the factor function associated with certain factor node f. $\sum_{\sim y_i}$ means to sum up the factor function over all variables except y_i. After LBP converges to a fixed point, the belief probability of each variable node y_i can be obtained as the product of all messages passing toward y_i.

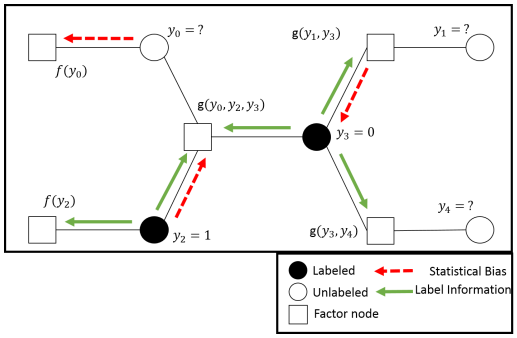


Figure 2: Modeling LBP with information diffusion: Streams of label information are diffused from labeled samples while flows of statistical bias are intrinsically distributed in the network.

2.2 Active Learning on Non-progressive Diffusion Model

We propose a novel idea of connecting the LBP process to the information diffusion model and develop an instantiation model for the active learning for networked data.

In the information diffusion theory, there are two state-of-the-art diffusion models: linear threshold model and independent cascaded model [13]. We consider the linear threshold model, where a sample will be activated as long as the number of its active neighbors exceeds its threshold. According to the diffusion process, the linear threshold model can be further classified into two categories: progressive model and non-progressive model.

In the progressive model, a node in the network can only be activated once and remains its status in the following diffusion process. More formally, let $f_\tau(v)$ denote the status of node v at time τ. If $f(v) = 1$ then v is activated, otherwise not. Let $NB(v)$ denote the neighbor set of v, the formal definition of f is as follows.

$$f_\tau(v) = \begin{cases} 1 & \text{if } \sum_{u \in NB(v)} f_{\tau-1}(u) \geq t(v) \text{ or } f_{\tau-1}(v) = 1 \\ 0 & \text{if } \sum_{u \in NB(v)} f_{\tau-1}(u) < t(v) \text{ and } f_{\tau-1}(v) = 0 \end{cases}$$

In the non-progressive model, a node in the network can reverse its status in both directions, i.e., either from active to inactive, or vice versa, depending on the status of its neighbors at the last time step. More formally,

$$f_\tau(v) = \begin{cases} 1 & \text{if } \sum_{u \in NB(v)} f_{\tau-1}(u) \geq t(v) \\ 0 & \text{if } \sum_{u \in NB(v)} f_{\tau-1}(u) < t(v) \end{cases}$$

Which model is more suitable for modeling the message passing process in an LBP? As illustrated in figure 2, we can imagine that there are two types of information diffusing across the social network. One is statistical bias, which may be caused by insufficiency of labeled data, imbalance of labeled data or redundancy of selected features. This kind of information is distributed randomly and intrinsically in the given network and may cause overfitting of the factor graph model. The other type is labeled information, which indicates the ground truth, counterbalances the statistical bias and thus enhances the capability of generalization of

factor graph model. Suppose statistical bias and labeling information are spreading across the network simultaneously, our goal is to maximize the influence of labeling information and minimize the side effects of statistical bias. In this perspective, as for an uncertain variable node, whose "belief" is around 0.5, it may be alternatively influenced by statistical bias and labeling information, and thus sway its "belief" probability between 0.49 and 0.51. For this reason, during LBP message passing, except the initially labeled nodes, a node predicted to have label y_i after iteration τ will be probably predicted to have label y_j after the next iteration $\tau + 1$. Therefore, apparently, it is more acceptable to select non-progressive model because a variable node in the factor graph, as we demonstrated above, can reverse its status in both directions.

Now we utilize non-progressive linear threshold model to instantiate the utility function in Eq. 1.

Problem 1. **Active Learning on Non-progressive Diffusion Model.** Given a factor graph model trained on the partially labeled network $G = (V_L, V_U, \mathbf{y}_L, E, \mathbf{X})$, the threshold values and uncertainty values for each sample, denoted as $\{t(v)|v \in V_L \cup V_U\}$ and $\{\mu(v)|v \in V_L \cup V_U\}$, and a budget k, we aim to query a subset of k unlabeled samples, such that the performance of factor graph model will be best improved. Suppose the uncertainty of variable v grows as $\mu(v)$ increases, the utility function can be defined as:

$$\max_{V_S \subseteq V_U} \{ \max_{V_T \subseteq V_U} |V_T|\}, \quad |V_S| \leq k$$

with the constraints:

$$f_0(v) = 1 \iff v \in V_S \tag{7}$$

$$\exists \tau_M \text{ s.t. } \forall v \in V_T \ \forall \tau > \tau_M \ f_\tau(v) = 1 \tag{8}$$

$$\forall v \in V_U \backslash V_T, \forall u \in V_T, \mu(v) \leq \mu(u) \tag{9}$$

$$f_\tau(v) = 1 \iff \sum_{u \in NB(v)} f_{\tau-1}(u) \geq t(v) \tag{10}$$

Here we interpret the constraints we made above. Constraint (7) indicates that we initially label all samples in V_S. By constraint (8) we require all samples in V_T to be eventually activated, because we assume that only an activated sample can be effectively influenced by label information. By constraint (9), we also require V_T is made up of top k uncertain nodes in V_U, instead of simply counting the number of eventually infected nodes. In this way of definition we are only interested in those samples that lie on the edge of right and wrong, which is the key point to the active learning problem, rather than treating every unlabeled sample equally. Constraint (10) indicates that we consider the active learning problem under non-progressive diffusion model. Because there may be more than one V_T satisfying the constraints, we select the maximum one in the utility function.

3. ALGORITHMS

In this section, we solve problem 1 by two steps. First we reduce the problem to Minimum Source Set problem. Second we give an approximate solution to Minimum Source Set problem.

3.1 Reduction

In problem 1, we fix the number of source set V_S and try to maximize the size of target set V_T. However, in this section, we define a Minimum Source Set problem where we fix the number of target set V_T and aim to minimize the number of queried samples.

Problem 2. **Minimum Source Set** Given a factor graph model trained on the networked data $G = (V_L, V_U, \mathbf{y}_L, E, \mathbf{X})$, the threshold values for each sample, denoted as $\{t(v)|v \in V_L \cup V_U\}$, and the target set $V_T \subseteq V_U$, we aim to find a minimal source set $V_S \subseteq V_U$ to eventually activate all samples in V_T, i.e.,

$$\min |V_S|$$

with the constraints that if $f_0(v) = 1$ for all $v \in V_S$, then $\exists \tau_M$ s.t. $f_t(v) = 1$ for all $v \in V_T$ and $t > \tau_M$. The samples in the graph are updated with the non-progressive rule:

$$f_\tau(v) = 1 \iff \sum_{u \in NB(v)} f_{\tau-1}(u) \geq t(v)$$

We are now to introduce the equivalence of problem 1 and problem 2.

Theorem 1. The problem of Active Learning on Non-progressive Diffusion Model(problem 1) and Minimum Source Set problem(problem 2) are equivalent.

PROOF. First, we introduce a reduction from problem 1 to problem 2. For problem 1, given $|V_S| = k$, we aim to find an instance of V_S to maximize $|V_T|$. We build a family of instances of problem 2 to solve problem 1. We enumerate $|V_T|$ from its maximum possible value $|V_U|$ and count down until $|V_T|$ can be achieved with $|V_S| \leq k$. We build an instance of problem 2 to find out the optimal solution $V_{S,\text{opt}}$ given $|V_T|$. If $|V_{S,\text{opt}}| \leq k$, then we can quit the algorithm and $V_{S,\text{opt}}$ is the optimal solution for the problem 1, with which the maximum size of $|V_T|$ is the one being enumerated, denoted as $|V_T|_m$.

Now we prove the reduction above by contradiction. Suppose that there exists another solution V_S^* such that $|V_S^*| \leq k$ and $|V_T^*| > |V_T|_m$, where V_T^* is the target set corresponding to V_S^*. Therefore, when we build an instance of problem 2 with $|V_T^*|$, the optimal solution $V_{S,\text{opt}}$ will satisfy $|V_{S,\text{opt}}| \leq |V_S^*| \leq k$. Thus, the algorithm will quit and $|V_T|_m$ will not be enumerated, which leads to contradiction. Therefore, the reduction above gives optimal solution to problem 1.

Next, we prove that problem 2 can be reduced to problem 1. For problem 2, given $|V_T| = l$, we aim to find a smallest source set V_S to eventually activate all samples in V_T. Similar to the reduction in the reverse direction, we can enumerate the value of $|V_S|$ in this case. Because we want to minimize $|V_S|$, we enumerate $|V_S|$ from 0 and count upwards until V_T can be activated by V_S. Again we leverage an instance of problem 1 to find the maximum size of target set $|V_T|_m$ with source set of size $|V_S|$. If $|V_T|_m \geq l$, we terminate the algorithm and the optimal solution is the one returned by problem 1, denoted as $V_{S,\text{opt}}$.

Now we prove the reduction above by contradiction. Suppose that there exists another solution V_S^* such that $|V_S^*| < |V_{S,\text{opt}}|$ and the size of eventually activated sample set $|V_T^*| \geq l$. When we enumerate $|V_S^*|$, we can leverage an instance of problem 1 to obtain a target set V_T with $|V_T| \geq l$. Therefore, $|V_{S,\text{opt}}|$ will not be enumerated, which contradicts the assumption.

It follows two problems are equivalent. □

Algorithm 1: MaxCo Algorithm part 1: Reduction

Input: $G = (V_U, V_L, \mathbf{y}_L, E, \mathbf{X})$, k, threshold function t
Output: V_S: the selected set of nodes to be labeled.

```
l ← min value for |V_T|
r ← max value for |V_T|
V_{S,opt} ← ∅
while l < r do
    |V_T| ← (l+r)/2
    V_S ← MinimumSourceSet(G, |V_T|)
    if |V_S| ≤ k then
        l ← |V_T| + 1
        V_{S,opt} ← V_S
    else
        r ← |V_T|
return V_{S,opt}
```

LEMMA 1. *For problem 2, if either $\mu(v_1) < \mu(v_2)$ or $\mu(v_2) < \mu(v_1)$ $\forall v_1, v_2 \in V_U \cup V_L$ and $v_1 \neq v_2$, $|V_{S,opt}|$ is monotonically non-decreasing with respect to $|V_T|$.*

PROOF. To prove the lemma, we only need to prove that for two target sets V_{T1} and V_{T2} and their optimal solutions V_{S1} and V_{S2}, if $|V_{T1}| > |V_{T2}|$, then we have $|V_{S1}| \geq |V_{S2}|$.

According to the definition of minimum source set problem, the target set should be top-l uncertain nodes. Because samples can be strictly ordered with respect to uncertainty, we can infer $V_{T2} \subset V_{T1}$ from the assumption that $|V_{T1}| > |V_{T2}|$.

Suppose $|V_{S1}| < |V_{S2}|$. Because V_{S1} can activate all samples in V_{T1} and $V_{T2} \subset V_{T1}$, V_{S1} can activate all samples in V_{T2} as well. It contradicts the assumption that V_{S2} is the optimal solution for V_{T2}. Therefore, $|V_{S1}| \geq |V_{S2}|$. □

In practice, we can force any two uncertainty values to be strictly ordered even if they are arithmetically equal. For example, we can assume that v_i is uncertain than v_j if $\mu(v_i) = \mu(v_j)$ but $i > j$. Therefore, because $|V_{S,\text{opt}}|$ is monotonically non-decreasing with respect to $|V_T|$, we can apply bisection method to optimize the reduction algorithm introduced in the proof of theorem 1. When reducing from problem 1 to problem 2, instead of enumerating the value of $|V_T|$ one by one, we can leverage bisection method to bisect the interval and select a subinterval recursively, which will largely speed up the algorithm from $\mathcal{O}(n)$ to $\mathcal{O}(\log n)$, where n indicates the size of domain of $|V_T|$. The complete procedure for reduction from problem 1 to problem 2 is illustrated in algorithm 1.

3.2 Threshold and Uncertainty

Because the problem of Active Learning on Non-progressive Diffusion Model can be reduced to Minimum Source Set problem in polynomial time, we now focus on solving the Minimum Source Set problem.

Before we solve Minimum Source Set problem, we need to give a definition of threshold function $t(v)$.

Threshold Definition. The threshold value of a sample reflects how sensitive it is to the status of its neighbors. In the factor graph model, if the "belief" probability of a sample v is quite close to $\frac{1}{|\mathcal{Y}|}$, i.e., variable node v is very uncertain, then it is easily activated by the messages passed from its neighbors. For this reason, samples with higher uncertainty should obtain a lower threshold. Moreover, because a sample with high degree will receive a great number of messages, which may contain both labeling information and statistical bias, its threshold should be relatively high. Considering both degree and uncertainty factors, we define the threshold function as follows.

$$t(v) = \min\{\lceil \eta(\mu\max - \mu(v))d(v) \rceil, d(v)\} \tag{11}$$

where $d(v)$ is the degree, $\mu\max$ is the maximum value of uncertainty measure and η is a global constant factor to adjust the distribution of thresholds. We apply ceiling operation to the expression to avoid some extreme cases where $t(v) = 0$. Also, of course, the value $t(v)$ should not be greater than $d(v)$, otherwise v will never be activated whatsoever.

Given Eq. 11, it is a nontrivial job to tune the parameter η because it varies in a wide range in different data sets. To have a unified representation, we further define η to be

$$\eta = \frac{\gamma}{\text{Avg}(\mu\max - \mu)} \tag{12}$$

where Avg means the average value over the given data set, and γ is a constant factor. Under the definition, we can instead tune γ between 0 and 1, which is relatively irrelevant to the specific data set.

Uncertainty Definition. Now what remains to be defined is an uncertainty function. There are several measures corresponding to the uncertainty of a variable node in a factor graph. One is to define it as the sum of difference between expected belief and calculated belief of each class [21].

$$\mu(v) = -\sum_{y \in \mathcal{Y}} \left| \mathcal{B}_v(y) - \frac{1}{|\mathcal{Y}|} \right| + 2\left(1 - \frac{1}{|\mathcal{Y}|}\right) \tag{13}$$

where $\mathcal{B}_v(y)$ is obtained by LBP, indicating the "belief" probability that v is classified to have label y. We add a constant term to ensure that $\mu(v)$ is always nonnegative.

Another extensively used measure of uncertainty is entropy(aka TTE in [22]). More formally,

$$\mu(v) = \sum_{y \in \mathcal{Y}} \mathcal{B}_v(y) \log \frac{1}{\mathcal{B}_v(y)} \tag{14}$$

We reserve two ways of definition here and judge them through experiments in § 5.

3.3 MinSS

After defining the threshold function and uncertainty function, in this section, we aim to solve Minimum Source Set problem.

First we claim that Minimum Source Set is an NP-hard problem, which will be proved in § 4. We design an approximate algorithm MinSS to find the source set V_S iteratively and greedily.

MinSS. The basic idea of MinSS is to find an extended target set V_τ. An extended target set should satisfy two constraints. First, V_τ is a superset of V_T. Second, V_τ can be eventually activated if we initially label a subset of V_τ.

Algorithm 2: MaxCo Algorithm part 2: MinSS

Input: $G = (V_U, V_L, \mathbf{y}_L, E, \mathbf{X})$, k, threshold function t
Output: V_S: the selected set of nodes to be labeled.

```
1  calculate B_v(y) and μ(v) by LBP
2  V_τ ← V_T ← {top k uncertain nodes in V_U}
3  V_S ← ∅
4  while V_S = ∅ do
5      V_P ← V_U \ V_τ
6      sort nodes in V_P in descending order of t(v) as
       v_1, v_2, ..., v_p
7      foreach v ∈ V_τ do
8          w(v) ← 0
9      for i ← 1 to p do
10         if w(u) < d(u) − t(u) ∀u ∈ NB(v_i) ∩ V_τ then
11             foreach u ∈ NB(v_i) ∩ V_τ do
12                 w(u) ← w(u) + 1
13             V_P ← V_P \ {v_i}
14     if V_P = ∅ then
15         sort nodes in V_τ in ascending order of d(v) as
           v_1, v_2, ..., v_m
16         foreach v ∈ V_τ do
17             w(v) ← 0
18         foreach i ← 1 to m do
19             if ∃u ∈ NB(v_i) ∩ V_τ st. w(u) = d(u) − t(u)
               then
20                 V_S ← V_S ∪ {v_i}
21             else
22                 foreach u ∈ NB(v_i) ∩ V_τ do
23                     w(u) ← w(u) + 1
24     V_τ ← V_τ ∪ V_P
25 return V_S
```

MinSS is an iterative algorithm. In each iteration, we greedily select samples to expand the size of V_τ until V_τ satisfies the constraints made above.

The procedure of MinSS is illustrated in Algorithm 2. We first select top k uncertain nodes in V_U to form the target set V_τ. Here we may define the uncertainty with either Eq. 13 or Eq. 14. Then we expand the target set V_τ iteratively. For each iteration, we sort all unselected samples $v \in V_P$ in descending order of threshold value because a smaller threshold value indicates less active neighbor samples required. From line 9 to line 13, we greedily remove samples from V_P with large threshold value while satisfying the constraint that each node $v \in V_\tau$ have at least $t(v)$ neighbors in $V_\tau \cap V_P$. In line 14, if $V_P = \emptyset$, then V_τ is an extended target set. This time we greedily remove samples from V_τ with small degree and all samples left form the source set V_S, which is done from line 15 to line 23. In algorithm 2, $w(v)$ is used to count the unselected neighbors of sample v.

So far, we solve the problem of Active Learning on Nonprogressive Diffusion Model by two steps. First, the problem is reduced to Minimum Source Set problem. Second, we solve Minimum Source Set problem by algorithm MinSS. We combine these two steps and refer to the algorithm as MaxCo.

4. THEORETICAL ANALYSIS

In this section, we theoretically analyze the problems defined in § 2 and the MinSS algorithm proposed in § 3.

4.1 NP-hardness

Now we introduce the tools for proofs in this section.

LEMMA 2. *Suppose $V_{S,opt}$ is an optimal solution for problem 2, if $\alpha|V_U| \leq |V_{S,opt}|$ for every bipartite graph H, then $\alpha|V_U| \leq |V_{S,opt}|$ for every graph G.*

LEMMA 3. *Minimum Source Set problem is NP-hard when $V_T = V_U$.*

The proofs of lemma 2 and lemma 3 could be found in [8]. It is trivial to show that Minimum Source Set problem is NP-hard.

LEMMA 4. *Minimum Source Set problem is NP-hard.*

PROOF. The theorem follows directly from lemma 3 because the problem in lemma 3 is a special case of Minimum Source Set problem. □

COROLLARY 1. *The problem of Active Learning on Nonprogressive Diffusion Model is NP-hard.*

PROOF. By lemma 4 and theorem 1, the conclusion follows. □

4.2 Convergence and Correctness

Now we analyze the convergence issue for MinSS algorithm.

LEMMA 5. ***Convergence.*** *The MinSS algorithm will converge within $\mathcal{O}(|V_U| - |V_T|)$ time.*

PROOF. Denote V_τ after each round of iteration as a sequence $V_{\tau 0}, V_{\tau 1}, ..., V_{\tau\gamma}$. For each iteration i, if $V_P = \emptyset$, the loop of iteration will be terminated since V_S will be a nonempty set. Therefore, if $i < \gamma$, then $V_P \neq \emptyset$. It follows that $|V_{\tau i+1}| > |V_{\tau i}|$ for $i \in \{0, 1, ..., \tau - 1\}$. That is, $|V_{\tau i}|$ is strictly monotonically increasing.

It is trivial to show that $|V_{\tau i}| \leq |V_U|$ and $|V_{\tau 0}| = |V_T|$. Therefore, the length of sequence $\{V_{\tau i}\}_{i=0}^{\gamma}$ is finite. More precisely, $|\gamma| \leq |V_U| - |V_T|$, which yields the conclusion. □

Then we formally prove that MinSS algorithm will return a feasible solution once it converges.

Theorem 2. **Correctness.** If MinSS algorithm converges, V_S is a feasible solution. That is, if we initially label all samples in V_S, there exists a τ_H such that $f_\tau(v) = 1$ for all $v \in V_T$ and $\tau \geq \tau_H$.

PROOF. The MinSS algorithm converges if and only if $V_P = \emptyset$. In addition, after each iteration, $v \in V_P$ iff $\exists u \in \mathrm{NB}(v) \cap V_\tau$ st. $w(u) \geq d(u) - t(u)$. At the end of the algorithm, because V_P is empty, we have $w(v) < d(v) - t(v)$ for all $v \in V_\tau$. That is,

$$|\mathrm{NB}(v) \cap V_\tau| \geq t(v) \text{ for all } v \in V_\tau$$

Now we prove $f_\xi(v) = 1$ for all $v \in V_\tau$ and $\xi \geq 1$ by induction. For $\xi = 1$, since $w(v) < d(v) - t(v)$ for all $v \in V_\tau$,

$\sum_{u\in \mathrm{NB}(v)} f_0(u) \geq t(v)$. Therefore, $f_1(v) = 1$ for all $v \in V_\tau$. For $\xi > 1$,

$$\sum_{u\in \mathrm{NB}(v)} f_{\xi-1}(u) \geq \sum_{u\in \mathrm{NB}(v)\cap V_\tau} f_{\xi-1}(u)$$

By induction hypothesis, $f_{\xi-1}(v) = 1$ for all $v \in V_\tau$. Therefore, for all $v \in V_\tau$,

$$\sum_{u\in \mathrm{NB}(v)} f_{\xi-1}(u) \geq |\mathrm{NB}(v) \cap V_\tau| \geq t(v)$$

Then we have $f_\xi(v) = 1$ for all $v \in V_\tau$. Because $V_T \subseteq V_\tau$, the conclusion follows. □

4.3 Approximation Ratio

Now we show a lower bound for the optimal solution to problem 2 when $V_T = V_U$.

Theorem 3. **Lower Bound.** Let $D(V) = \sum_{v\in V} d(v)$, $T(V) = \sum_{v\in V} t(v)$, and suppose $t(v) \leq \beta d(v)$ for all $v \in V$. If $2T(V_U) - D(V_U) > 0$ and $V_T = V_U$, we have an lower bound for optimal solution $\left|V_{S,\mathrm{opt}}\right|$ to problem 2.

$$\left|V_{S,\mathrm{opt}}\right| \geq \frac{2T(V_U) - D(V_U)}{\beta\Delta} \tag{15}$$

The proof of Theorem 3 is given in the appendix. Because MinSS is an approximate algorithm, we give an upper bound for it as follows. We denote Δ as the maximum degree among nodes in the graph G.

Theorem 4. **Upper Bound.** Suppose $t(v) \leq \beta d(v)$ for all $v \in V$, we can derive an upper bound for MinSS algorithm.

$$|V_S| \leq \frac{\beta\Delta}{1-\beta+\beta\Delta}|V_U|$$

The detailed proof of Theorem 4 is given in the appendix. With an upper bound and a lower bound, we can prove an approximation ratio when $V_T = V_U$.

COROLLARY 2. ***Approximation Ratio.*** *Let $V_{S,g}$ denote the solution given by MinSS algorithm, $V_{S,opt}$ represent the optimal solution and Δ be the maximum degree in the graph. Suppose $t(v) \leq \beta d(v)$ for all $v \in V$, if $V_T = V_U$ and $2T(V_U) > D(V_U)$, we have*

$$\frac{|V_{S,g}|}{\left|V_{S,opt}\right|} \leq \frac{(\beta\Delta)^2}{(1-\beta+\beta\Delta)\cdot Avg[2t(v)-d(v)]} \tag{16}$$

where Avg[.] represents the expectation over all samples in the network.

PROOF. By Theorem 3, we have

$$\left|V_{S,\mathrm{opt}}\right| \geq \frac{2T(V_U) - D(V_U)}{\beta\Delta} = \frac{\mathrm{Avg}[2t(v)-d(v)]}{\beta\Delta}|V_U| \tag{17}$$

By Theorem 4, we have

$$|V_{S,g}| \leq \frac{\beta\Delta}{1-\beta+\beta\Delta}|V_U| \tag{18}$$

Dividing (18) by (17) yields the conclusion. □

The approximation ratio given in equation (16) depends on two variables β and $\mathrm{Avg}[2t(v) - d(v)]$. Both of them will be affected if we adjust the value of η given in equation (11). To optimize the approximation ratio, we need to tune β as small as possible and set $\mathrm{Avg}[2t(v)-d(v)]$ as large as possible. However, it is contradictory. Suppose we tune up the value of η, then according to definition of threshold value, in general, β and $\mathrm{Avg}[2(v) - d(v)]$ will both be tuned up. And for another thing if we tune down η, both β and $\mathrm{Avg}[2t(v) - d(v)]$ will be tuned down as well. Thus, for practical application, we need to carefully choose the value of η such that we can balance the effects of β and $\mathrm{Avg}[2t(v)-d(v)]$. This issue will be further discussed in § 5 by experiments.

5. EXPERIMENTAL RESULTS

The proposed active learning framework is general and can be applied to arbitrary networked data. In this section, we evaluate the proposed algorithm (MaxCo) and compare with existing methods. All codes and datasets used in this paper can be found at `http://arnetminer.org/maxco/`.

5.1 Datasets and Comparison Methods

We consider four social network datasets in our evaluation: Coauthor, Slashdot, Mobile and Enron. In all these datasets, we aim to infer the type of social relationships in the different networks. Regarding the partially labeled factor graph model, we use the implementation from [28, 31].[1] In the factor graph model, we view each relationship as a variable node and define factor functions according to the specific properties of each network.

- Coauthor. The dataset is a subgraph extracted from ArnetMiner [27]. In this dataset, we aim to infer advisor-advisee relationship from the given network. The factor graph built upon this dataset consists of 6096 variable nodes and 24468 factor nodes.
- Slashdot. The dataset is crawled from Slashdot website. We try to infer friendship on this network. The factor graph built upon this network contains 370 variable nodes and 1686 factor nodes.
- Mobile. This dataset contains logs of call, blue tooth scanning and location collected by mobile applications on 107 phones in a span of 10 months. We aim to infer friendship on this dataset. The factor graph built upon the data consists of 314 variable nodes and 513 factor nodes.
- Enron. The dataset consists of 136,329 emails among 151 Enron employees. We aim to infer manager subordinate relationship from the network. The factor graph model built has 100 variable nodes and 236 factor nodes.

We the compare our algorithm (MaxCo) with the following methods for the problem of batch mode active learning for networked data.

- Random(RAN). In this method, each time we randomly select given number of samples to label.

[1]The source code is available at http://keg.cs.tsinghua.edu.cn/jietang/software/OpenCRF_PartiallyLabeledFGM.rar

Table 1: Factor Graph Size

Data	#Variable Node	#Factor Node
Coauthor	6,096	24,468
Slashdot	370	1,686
Mobile	314	513
Enron	100	236

Table 2: Average Accuracy(%)

Data	IMS	MaxCo	MU	RAN	BMAL
Coauthor	74.24	**82.70**	46.99	79.72	44.92
Enron	71.17	85.33	**85.67**	83.67	84.67
Slashdot	66.95	**69.62**	66.11	67.00	66.16
Mobile	67.83	**76.15**	59.68	63.73	55.86

- Maximum Uncertainty (MU). This method greedily selects samples with great entropy(Eq. 14).
- Batch Mode Active Learning (BMAL). This method is proposed by [23], which aims to maximize the following quality function of selected sample set V_S.

$$Q(V_S) = \alpha C(V_S) + (1-\alpha)H(V_S), 0 \leq \alpha \leq 1$$

where the definition of $H(V_S)$ and $C(V_S)$ is as follows.

$$H(V_S) = \sum_{i \in V_S} H(i)$$

$$C(V_S) = \sum_{i \in V_U} (H(i))^{\beta} \left(\max_{j \in V_L \cup V_S} w_{ij} \right)^{1-\beta}$$

Here $H(i)$ is the entropy of a variable node i and β is a constant parameter; w_{ij} denotes the similarity between variable i and j, and can be calculated by $e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}}$, where $\mathbf{x}$ represents an attribute vector for each variable.

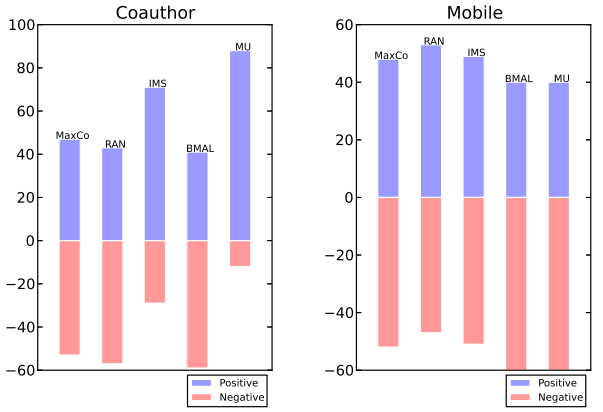
- Influence Maximization Selection (IMS). IMS is proposed by [31], which also utilizes information diffusion model to solve the active learning problem. However, it is based on progressive diffusion model.
- Maximum Coverage (MaxCo). We use entropy to measure uncertainty and empirically set $\gamma = 0.7$(Eq. 12).

For each dataset, we randomly label 10 samples to form $\mathbf{y}_L$ at the beginning. Then we iteratively apply the active learning algorithm, by selecting 10 samples to query each time. After each round, we train the factor graph to test the accuracy and F1 score.

5.2 Performance Analysis

Table 3: Average F1-score(%)

Data	IMS	MaxCo	MU	RAN	BMAL
Coauthor	69.55	**76.15**	59.68	63.73	55.86
Enron	79.50	**87.94**	86.30	85.85	85.59
Slashdot	76.90	**80.04**	77.85	77.60	77.74
Mobile	69.51	**75.31**	51.23	50.17	53.57



(a) Coauthor: #Label=100 (b) Mobile: #label=100

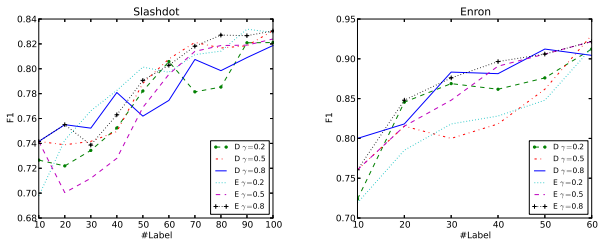
Figure 3: Labeled Data Balance: Blue bars above 0 represent the number of positive selected samples and red bars below 0 indicate the number of negative selected samples with different active learning strategies.

Figure 1 shows F1-score and accuracy performance of each algorithm on each dataset. We also calculate the average accuracy performance and F1 score for all selection strategies, which is shown in Table 2 and Table 3 respectively.

Performance Comparison. According to Figure 1 and Table 2 and 3, we see that MaxCo significantly outperforms other competing methods on the four datasets. In Coauthor, MaxCo is better than the random selection method by 2.98% and 12.42% improvements in terms of average accuracy and F1 score. Other methods such as MU and BMAL seems to amplify the side effects of label imbalance, which will be discussed later. In Enron, MaxCo is 0.43% worse than MU by average accuracy, but 1.64% better in terms of average F1. These two methods strongly outperform other methods. In Slashdot and Mobile, MaxCo achieves outstanding performance while other methods perform relatively closely to each other.

Imbalance of Labeled Data. From the results, we found a phenomenon: for some active learning strategies, performance will decrease as the number of labels grows. By further investigation, we discover that the phenomenon is related to labeled data imbalance. From figure 1 we can see that, Coauthor and Mobile datasets are relatively more sensitive to the balance of labeled data. For these two datasets, the curves representing MU and BMAL turn down sharply as the number of labeled instances increases. When the number of labeled data comes to 100, the F1 score falls to around 50% in the Coauthor data and 30% in the Mobile data, the accuracy performance also suffers sharp decrease. We observe the ratio of positive and negative instances and find that the ineffectiveness of these two models in this case is correlated to imbalance of labeled data.

Figure 3 plots the number of positive and negative labeled samples when the number of labeled samples is 100. Figure 3(a) is for the Coauthor dataset and figure 3(b) is for the Mobile dataset. On the Coauthor dataset, IMS and MU tend to label positive instances while BMAL labels significantly more negative instances. On the Mobile dataset, also, BMAL and MU suffer from labeled data imbalance,



(a) Slashdot (b) Enron

Figure 4: Parameter Sensitivity: D denotes distance measure of uncertainty and E represents entropy measure; γ is the constant factor in threshold function.

where negative instances take up to 60 percent of all labeled samples. The results show that for some datasets, the performance of factor graph model will be influenced by the balance of labeled samples. The results also demonstrate that MaxCo can produce relatively balanced results.

Significance Test. We perform significance test for the results of the comparison methods. Pairing the measure scores(F1 or accuracy) of two models with the same number of labeled instances on a dataset, we assume that the difference of the two models is random and symmetric around the median. Therefore, we can perform Wilcoxon signed-rank test [24] to demonstrate the significance level of the difference between the two models. The result shows that p-value is less than 0.01, which indicates that the improvements of MaxCo over the competing methods are statistically significant.

Parameter Sensitivity. We further study the parameter sensitivity of γ in MaxCo. We also consider the effects of uncertainty definition. According to § 3, we compare distance measure (Eq. 13) with entropy measure (Eq. 14). We repeat our experiment with different ways of threshold definition and uncertainty definition. The results are plotted in Figure 4. We can see that MaxCo is insensitive to the threshold parameter and uncertainty definition. Entropy measure is slightly better than distance measure, with a relatively larger γ.

6. RELATED WORK

Active learning is a very important topic in the study of social network and web mining because of exponentially growing size of data and high cost of data labeling. Settles et al. [22] surveyed query selection strategies for sequence models and proposed novel algorithms. There have been several works designing active learning algorithms to specific problems. For example, Arasu et al. [1] present novel algorithms for the problem of record matching packages. Hoi et al. [12] studied active learning on text categorization problem. Different from existing methods, we propose a general framework, which can be applied to different problems. There are also several works based on specific models. Martinez et al. [16] formulated the active learning problem under for the conditional random field (CRF) model. Golovin et al. [10] developed a greedy algorithm for Bayesian active learning with noisy observations. A similar work [31] by Zhuang et al. also studied active learning problem on factor graph model. However, they utilize progressive information diffusion to solve the problem, which is demonstrated to be less effective than non-progressive one in this paper. In [23], Shi et al. proposed a general framework on batch mode active learning. They used three criteria for instance selection. We also compare to their algorithm in §5. Also, extensive literature focused on active learning on social network [15, 14, 2, 20].

Our work is also related to information diffusion model. In [13], Kempe et al. solved the influence maximization problem on progressive diffusion model and show a reduction to non-progressive one. However, their definition of non-progressive model cannot be applied to the active learning problem. There are several works studying the spread of social influence with various propagation models [9, 18, 30, 6]. Besides, the diffusion models are widely used in real-world applications such as viral marketing [19, 7]. Progressive diffusion models have been extensively studied in the literature [25, 4, 11, 3, 26]. Fazli et al. [8] proposed a greedy algorithm for non-progressive model and proved approximation ratio on power law graph. However, their works cannot be directly applied to active learning problem for the network data.

7. CONCLUSION

In this paper, we study the problem of batch mode active learning for networked data, which aims to query k unlabeled samples in a network such that we can achieve best performance improvement. We utilize factor graph model as our basic framework so as to leverage link formation of networked data. We leverage Loopy Belief Propagation to learn the parameter in factor graph model. We propose a novel idea of connecting the graphical model to the information diffusion process. Therefore, we precisely instantiate the active learning problem on a non-progressive diffusion model.

We solve the problem of active learning on non-progressive diffusion model by MaxCo, which includes two steps. First we prove a reduction to Minimum Source Set problem and then we propose an iterative greedy algorithm MinSS to solve the Minimum Source Set problem. We theoretically show the NP-hardness of our problem, analyze the convergence and correctness of MinSS, and finally provide approximation guarantees for our algorithm with an upper bound and a lower bound. We empirically evaluate MaxCo algorithm in comparison with several baseline methods on several datasets. The experimental results demonstrate that our approach significantly outperforms the competing methods.

Acknowledgements. The work is supported by National Basic Research Program of China (No. 2011CB302302) and Natural Science Foundation of China (No. 61222212, No. 61073073), and a research fund supported by Huawei Inc.

8. REFERENCES

[1] A. Arasu, M. Götz, and R. Kaushik. On active learning of record matching packages. In *SIGMOD'10*, pages 783–794, 2010.

[2] M. Bilgic, L. Mihalkova, and L. Getoor. Active learning for networked data. *ICML*, pages 79–86, 2010.

[3] C. Chang and Y. Lyuu. Spreading messages. *Theor Comput Sci*, 410:2714–2724, 2009.

[4] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD'10*, pages 1029–1038, 2010.

[5] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *J. Artif. Int. Res.*, 4(1):129–145, Mar. 1996.
[6] Z. Dezsö and A. Barabási. Halting viruses in scale-free networks. *Phys Rev*, 2002.
[7] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD'01*, pages 57–66, 2001.
[8] M. Fazli, M. Ghodsi, J. Habibi, P. J. Khalilabadi, V.Mirrokni, and S. S. Sadeghabad. On the non-progressive spread of influence through social networks. *LATIN 2012: Theoretical Informatics*, 7256:315–326, 2012.
[9] L. Freeman. *The development of social network analysis*. Empirical Press Vancouver, British Columbia, 2004.
[10] D. Golovin, A. Krause, and D. Ray. Near-optimal bayesian active learning with noisy observations. *CoRR*, 2010.
[11] A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *WSDM'10*, pages 241–250, 2010.
[12] S. C. H. Hoi, R. Jin, and M. R. Lyu. Large-scale text categorization by batch mode active learning. In *WWW'06*, pages 633–642, 2006.
[13] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD'03*, pages 137–146, 2003.
[14] M. Kimura, K. Satio, R. Nakano, and H. Motoda. Extracting influential nodes on a social network for information diffusion. *DMKD*, 20:70–97, 2010.
[15] A. Kuwadekar and J. Neville. Relational active learning for joint collective classification models. In *ICML'11*, pages 385–392, 2011.
[16] O. Martinez and G. Tsechpenakis. Integration of active learning in a collaborative crf. In *CVPRW'08*, pages 1–8, 2008.
[17] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *UAI'99*, pages 467–475, 1999.
[18] R. Pastor-Satorras and A. Vespignani. Epidemic spreading in scale-free networks. *Phys Rev*, 65, 2002.
[19] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD'02*, pages 61–70, 2002.
[20] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. *ICML*, pages 441–448, 2001.
[21] T. Scheffer, C. Decomain, and S. Wrobel. Active hidden markov models for information extraction. In *CAIDA'01*, pages 309–318, 2001.
[22] B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In *EMNLP*, pages 1070–1079, 2008.
[23] L. Shi, Y. Zhao, and J. Tang. Batch mode active learning for networked data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2011.
[24] Siegel and Sidney. *Non-parametric statistics for the behavioral sciences*. New York: McGraw-Hill, 1956.
[25] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *KDD'09*, pages 807–816, 2009.
[26] J. Tang, S. Wu, and J. Sun. Confluence: Conformity influence in large social networks. In *KDD'13*, pages 347–355, 2013.
[27] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: Extraction and mining of academic social networks. In *KDD'08*, pages 990–998, 2008.
[28] W. Tang, H. Zhuang, and J. Tang. Learning to infer social ties in large networks. In *ECML/PKDD'11*, pages 381–397, 2011.
[29] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, Jan. 2008.
[30] D. Wilson. Levels of selection: An alternative to individualism in biology and the human sciences. *Soc Networks*, 11:257–272, 1989.
[31] H. Zhuang, J. Tang, W. Tang, T. Lou, A. Chin, and X. Wang. Actively learning to infer social ties. *Data Mining and Knowledge Discovery*, 25(2):270–297, 2012.

APPENDIX

Proof of Theorem 3.

PROOF. First, suppose $G = (X, Y)$ is a bipartite graph. Let e_A denote the number of edges in A and e_{AB} denote the number of edges across A and B. Let $B_X = V_S \cap X$, $B_Y = V_S \cap Y$ and $W = V_U \backslash V_S$. Following the lemma in [8], we have

$$e_{WB_X} + e_W \leq \sum_{v \in B_X \cup W} (d(v) - t(v))$$

$$e_{WB_Y} + e_W \leq \sum_{v \in B_Y \cup W} (d(v) - t(v))$$

Because $\sum_{v \in W} d(v) = 2e_W + e_{WB_X} + e_{WB_Y}$, it follows

$$\sum_{v \in W} d(v) \leq \sum v \in V_U (d(v) - t(v)) + \sum_{v \in W} (d(v) - t(v))$$

Therefore,

$$T(W) \leq D(V_U) - T(V_U)$$

Because $T(V_S) = T(V_U) - T(W)$, we have

$$T(V_S) \geq T(V_U) - (D(V_U) - T(V_U)) = 2T(V_U) - D(V_U)$$

Because $t(v) \leq \beta d(v)$ for each $v \in V$, we can derive

$$T(V_S) \leq \beta D(V_S) \leq \beta \Delta |V_S|$$

Therefore, it yields

$$\left|V_{S,\text{opt}}\right| \geq \frac{2T(V_U) - D(V_U)}{\beta \Delta}$$

By Lemma 2, the inequality holds as well for any general graph G. □

Proof of Theorem 4.

PROOF. Let $Q = \{v | w(v) = d(v) - t(v) \wedge v \in V_U\}$, and $W = V_U \backslash V_S$. By definition we have,

$$\sum_{v \in Q} (d(v) - t(v)) \leq \sum_{v \in W} d(v)$$

Therefore,

$$(1 - \beta) \sum_{v \in Q} d(v) \leq \sum_{v \in W} d(v)$$

Referring to the procedure of MinSS algorithm, a sample v is inserted into V_S if and only if $\text{NB}(v) \cap Q \neq \emptyset$. Because for $v \in Q$, $w(v) = d(v) - t(v)$, we have

$$|V_S| \leq \sum_{v \in Q} t(v) \leq \beta \sum_{v \in Q} d(v) \leq \frac{\beta}{1 - \beta} \sum_{v \in W} d(v) \leq \frac{\beta}{1 - \beta} \Delta |W|$$

Because $|W| = |V_U| - |V_S|$, it follows,

$$|V_S| \leq \frac{\beta \Delta}{1 - \beta + \beta \Delta} |V_U| \quad \square$$