IEICE TRANS. INF. & SYST., VOL.E99–D, NO.10 OCTOBER 2016

PAPER

Latent Attribute Inference of Users in Social Media with Very Small Labeled Dataset*

Ding XIAO[†a)], ***Member***, **Rui WANG**[†], ***and*** **Lingling WU**[†], ***Nonmembers***

SUMMARY With the surge of social media platform, users' profile information become treasure to enhance social network services. However, attributes information of most users are not complete, thus it is important to infer latent attributes of users. Contemporary attribute inference methods have a basic assumption that there are enough labeled data to train a model. However, in social media, it is very expensive and difficult to label a large amount of data. In this paper, we study the latent attribute inference problem with very small labeled data and propose the SRW-COND solution. In order to solve the difficulty of small labeled data, SRW-COND firstly extends labeled data with a simple but effective greedy algorithm. Then SRW-COND employs a supervised random walk process to effectively utilize the known attributes information and link structure of users. Experiments on two real datasets illustrate the effectiveness of SRW-COND.

key words: *attribute inference, social network, supervised random walk, community detection*

1. Introduction

Recently, there is a surge of social media, which helps people to create, share, and exchange information and ideas in virtual communities and networks. Many social media platforms have been popular in our daily life, such as Facebook, Twitter and Weibo. Although users in these platforms are required to fill their personal information out (e.g., gender, affiliation and role), a lot of users do not fill out their profile attributes or only some of them. However, the attributes of users are very important in many applications. For example, sales person find the target customer through the affiliation attribute of users [1], and the identity of users can be determined by their attributes [2].

Many efforts have been made to infer the latent attributes of users [3]–[7]. Some of them consider it as a classification problem and train a classifier on labeled dataset. Some of them utilize the network structure to infer attribute according to the principle of homophily or social influence [8]. [9] utilize locations of a user's friends to predict the user's geographical location. These methods all have a basic assumption that we have a big enough labeled dataset to train a model. However, this is not always the case, especially in social media. The user information in social media is very sparse and highly fragmented. It is not only expensive but also difficult to label these data even by manual work. For example, there are only 179 authenticated users among more than 30K unlabeled users in China Mobile in Beijing. Can we infer the attributes of unlabeled users from such a very small number of labeled users?

In this paper, we study the attribute inference problem with very small labeled data and propose the Supervised Random Walk based on CONDuctance method (SRW-COND). The basic idea of SRW-COND is that it extends the small number of labeled data (i.e., seed nodes) with a simple but effective method, and then infers latent features through utilizing the structure and attributes information of users. In the labeled data extension phase, SRW-COND employs a greedy algorithm to automatically extend dense seed community with high accuracy and low time cost. In the attribute inference phase, SRW-COND propagates the seed information along link structure with varying propagating probability decided by node attributes information. Experiments on two real datasets verify that SRW-COND can achieve better performances in inferring users' latent attributes under very small labeled data, compared with other well established methods.

The rest of the paper is organized as follows. Section 2 briefly surveys related work. In Sect. 3, we describe the model of SRW-COND. We proceed by describing experimental evaluation in Sect. 4 and conclude in Sect. 5.

2. Related Work

Approaches for attribute inference can be roughly grouped into the following two categories.

Feature-based methods extract feature vector from node content or network structure to train a classifier, e.g., SVM [3], [10], GBDT [11], Naive Bayes [12]. All these work concentrate on feature extraction and their main contributions lie in finding the effective features. For example, Pennacchiotti et al. used available and unstructured online data generated by individuals to infer demographic attributes such as age, ethnicity, and political orientation for individual and groups of users [11]. Zamal et al. extended n-gram features to user' neighborhood and assessed the impact of different subsets of neighborhood to the inference of three attributes: gender, political orientation, and age [3]. In this kind of methods, they usually need a great amount of

Manuscript received February 1, 2016.
Manuscript revised May 27, 2016.
Manuscript publicized July 20, 2016.

[†]The authors are with the Beijing University of Posts and Telecommunications, China.

[*]This work is supported in part by National Key Basic Research and Department (973) Program of China (No. 2013CB329606), and the National Natural Science Foundation of China (No. 61375058), and the Co-construction Project of Beijing Municipal Commission of Education.

a) E-mail: dxiao@bupt.edu.cn

DOI: 10.1587/transinf.2016EDP7049

labeled data.

Link-based methods utilize network structure to determine the unknown attributes of users. Roth et al. inferred the latent attributes of user u's friends by ranking the users in u's egocentric network according to an interaction-based metric [13]. Zheleva et al. proposed an entropy-based method to reduce the large number of potential groups in order to improve the attribute accuracy [14]. Mislove et al. studied user attribute inference in university social networks by applying community detection [5]. These methods may have less satisfying performances because of only considering the structural information but ignoring easily available attributes information of users.

Backstorm and Leskovec proposed a Supervised Random Walk (SRW) framework to predict links in social networks [15]. SRW naturally combines network structure and link attributes and achieves satisfying performances in link prediction. But its goal is to predict links which differs from our attribute inference problem and there is a limited-labeled-data constraint in our problem setting. Recently, Zeng et al. studied the user's affiliation information inference problem and developed a supervised label propagation model which naturally incorporates the rich features of social activities among users [1]. Different from their method, our method is more generally applied on any user attribute inference and it focuses more on the difficulty of the lack of labeled data.

3. Problem Definition and Method

In this section, we first present the problem formulation, and then introduce SRW-COND.

3.1 Problem Definition

This study tries to discover the missing value of a target attribute of users. Without loss of generality, we infer whether users have a certain value on the target attribute. In the problem setting, we know the social network and some other attributes of users (note that these attribute values may be incomplete). The users having the certain value on the target attribute are positive examples. In real applications, the amount of these labeled data is usually very small. The problem is how can we infer whether other users have the same certain value on the target attribute with these very small labeled data. Figure 1 illustrates a toy example of the attribute inference problem. In Fig. 1, the colored boxes represent the target attributes. And the ellipses represent the values of attributes. The symbols (+), (−) and ? in the ellipses denote the positive, negative and unknown examples respectively.

Next, we formulate the attribute inference problem. Given a social graph $G = (V, E, X)$, where V is the set of nodes, E is the set of edges, and X is an $|V| \times d$ attribute matrix associated with nodes in V. x_{ij} in X denotes the value of the j^{th} attribute of node i. We only know that some nodes have certain value of the target attribute (i.e., positive examples, called V^{L_P}) and some nodes have not (i.e., negative examples, called V^{L_N}). Attribute values of other nodes are unknown (called V^U). Notice that $V^L = V^{L_P} + V^{L_N}$. It is obvious that $V = V^L + V^U$. There is a constraint that $|V^L| << |V^U|$. The problem we are trying to solve here is that given the network and V^L, we infer users in V^U whether they have the same value of the target attribute with users in V^{L_P}. However, in real applications, it is much more easier to gain positive examples than negative examples. For example, we gain positive examples in Sina Weibo dataset simply by specifying the users' affiliation to be "China Mobile" and then crawling through the Sina Weibo API. So we focus on the case where $V^L = V^{L_P}$ for the rest of the paper.

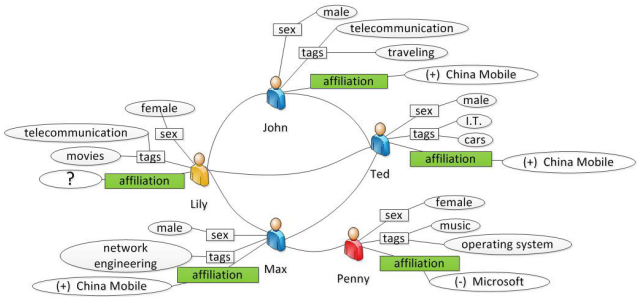


Fig. 1 A toy example of the attribute inference problem

3.2 Basic Idea

For the attribute inference problem, a simple solution is to consider it as a binary classification problem and train a classifier with the labeled data. However, the labeled dataset is very small, which is not enough to effectively train a classifier. In order to tackle the lack of labeled data, a basic idea is to extend the positive examples with a simple but effective method. Here we design a greedy algorithm to automatically extend dense seed set (i.e., positive examples) with high accuracy and low time cost. We may use seed set to refer to the set of positive examples for the rest of the paper.

For the extended positive examples, we can still apply the binary classification model, but it also needs to label the negative examples. Moreover, it is more promising to improve the prediction precision through effectively utilizing the social network structure and attributes information of users together. Inspired by the idea of supervised random walk [15], we propose a random walk based method to propagate the seed information along link structure with varying propagating probability which decided by attributes information of nodes.

3.3 Extension of Labeled Data

As a basic step of SRW-COND, the extension method needs to satisfy the following two requirements.

- It is simple. As a preprocessing step, the extension process cannot require much time and space cost.
- It is accurate. In order to avoid the cascade propagation

of wrong seeds information, the extended seeds should not be many but very accurate. That is, the extension process pays more attention to the *precision*, not the *recall*.

Inspired by the community based attribute inference [5], we design a greedy algorithm called Ave-Cond. to extend positive examples to form a dense community. That is, we add unlabeled nodes into the seed set to make their connection denser. The widely used *Conductance* C [16], shown in Eq. (1), is applied to evaluate the density of communities. For each step, an unlabeled node with the maximal increase of *Conductance* is added to the seed set. In order to prevent from adding negative examples, we design an adaptive and strict stop criterion, as shown in Eq. (2). With the increase of seed nodes, the *Conductance* increases, while the increase rate drops. Equation (2) evaluates the average increase rate. If the average increase rate of the last half phase drop too large (drop half), the node addition will be halted. We use V^L_{ext} to denote set A when the extension is over, that is, the extended set. The extension process of labeled data is shown in Algorithm 1.

$$C(A,B) = \frac{|E_{AB}|}{min(|E_A|,|E_B|)} \tag{1}$$

$$\frac{C^{(t)} - C^{(\lfloor t/2 \rfloor)}}{t - \lfloor t/2 \rfloor} < \frac{1}{2} \times \frac{C^{(\lfloor t/2 \rfloor)} - C^{(0)}}{\lfloor t/2 \rfloor} \tag{2}$$

$$C_{norm}(A,B) = \frac{|E_{AA}|}{|E_{AA}| + |E_{AB}|} - \frac{|E_A| \cdot |E_A|}{|E_A| \cdot |E_A| + |E_A| \cdot |E_B|} \tag{3}$$

In this paper, A is the extending labeled set and A is initialized with positive examples V^{L_P}. $B = V \backslash A$. $|E_{AB}|$ is the number of edges between A and B. $|E_A| = |E_{AB}| + |E_{AA}|$ and $|E_{AA}|$ is the number of edges within A. $|E_B| = |E_{AB}| + |E_{BB}|$ and $|E_{BB}|$ is the number of edges within B. t denotes the iteration times. $C^{(t)}$, $C^{(\lfloor t/2 \rfloor)}$ and C^0 represents the value of *Conductance* after t, $\lfloor t/2 \rfloor$ and 0 iterations respectively. The symbol $\lfloor\ \rfloor$ denotes round down operation.

The similar idea has been applied in user attribute inference [5], whose idea is to group users with the same attribute into a dense community through the *Normalized Conductance* criterion. The *Normalized Conductance* is shown in Eq. (3). In order to satisfy the high precision requirement, our method is mainly different from their algorithm in the following two aspects.

- We employ the *Conductance* criterion instead of the *Normalized Conductance* criterion for node selection.
- We use the new stop criterion as shown in Eq. (2).

The experiments show that our method can extend smaller positive examples with much higher accuracy. We think the reason lies in that we use different criteria for the node selection and the halt condition respectively (i.e. Eq. (1) and Eq. (2)). Although *Normalized Conductance* helps to prevent oversize communities through introducing a penalty factor, it may sacrifice the community structure characteristics. Note that we only need to find a small number of seed nodes, so in this condition, *Conductance* is better to find the good nodes with significant community structure characteristics, compared to *Normalized Conductance*. In addition, we employ a stricter halt condition (i.e. Eq. (2)), which only extends a small number of accurate seed nodes. The details can be seen in Sect. 4.3.2.

Algorithm 1 Extension of Labeled Data

Input: seed set V^L, network graph $G = (V, E, X)$
Output: extended seed set V^L_{ext}
$A \leftarrow V^L$
$B \leftarrow V \backslash A$
while equation (2) is not satisfied **do**
  $argmax_b\ C(A \cup \{b\}, B \setminus \{b\})$
  $B \leftarrow B \setminus \{b\}$
  $A \leftarrow A \cup \{b\}$
end while
$V^L_{ext} \leftarrow A$

3.4 Inference of Missing Attributes

According to homophily theory [17], we know that people usually form a community with others who are similar to them. So one way to infer the missing attributes is to find users who are similar to the seed set users with expectation that they would share the same target attribute value. Random walk with restart at extended seeds set V^L_{ext} is able to measure the similarity of other nodes with extended seed set V^L_{ext}, and the stationary distribution of the walk process assigns each node a score (i.e. PageRank score p^T) denoting the closeness. So we can sort the nodes according to the descending order of the PageRank score, and the top k users are inferred to have the same target attribute value with seed users. The walk process could be written as Eq. (4):

$$p^T = (1-\alpha) \cdot p^T Q + \alpha \cdot 1(v \in V^L_{ext}) \tag{4}$$

α is the restart probability. p^T is a vector of visiting probabilities of all nodes. Q is the transition probability matrix of nodes in the graph.

The random walk with restart takes advantage of the network structure, but ignores the impact of user attributes. In order to achieve better performances, we need to make full use of user attributes and network structure together. Inspired by Backstorm and Leskvec's work [15], the walk process has a varying walk probability associated with node attributes, instead of a stationary walk probability decided by node degree (which is a measurement completely decided by network structure). Following this idea, we redesign the transition matrix Q. First, we construct the feature vector $X_e(u,v)$ of edge (u,v) from node attributes. For example, in Sina Weibo, we can construct edge features like these, how many times user u has @ user v from users' tweets, and how many common tags from users' tag attribute. For each edge feature, we assign a feature weight to denote its importance.

Then, the weight a_{uv} of edge (u, v) can be represented as a function of the feature vector $X_e(u, v)$ and the feature weight vector ω as follows: $a_{uv} = f_\omega(\omega^T X_e(u, v))$. In this paper, f_ω is the Sigmoid function as in Eq. (5):

$$f_\omega(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

And thus we can define the following transition probability matrix as in Eq. (6):

$$Q_{uv} = \begin{cases} \frac{a_{uv}}{\sum_{i \in N(u)} a_{ui}} & (u, v) \in E \\ 0 & otherwise \end{cases} \tag{6}$$

Q_{uv} is the u^{th} row and v^{th} column item in transition matrix. a_{uv} and a_{ui} is the edge weight of edge (u, v) and edge (u, i) respectively. $N(u)$ denotes the neighbors of user u. According to the above representation, the transition matrix is a function of feature weight vector ω (i.e., $Q(\omega)$), instead of a constant decided by network structure. As shown in Eq. (4), the rank result is decided by transition matrix Q. Thus, we can adjust the transition matrix by changing the feature weight vector ω in order to rank the nodes in extended seed set V^L_{ext} higher. To achieve that, we can bias the random walk probability on those seed nodes by learning the parameters of function $f_\omega(\omega^T X_e(u, v))$. Because the random walker is more likely to traverse edges with high strength and the random walker jumps back at V^L_{ext} with probability of α, the nodes connected with V^L_{ext} through edges of high strength would rank higher. In this way, we can naturally combine the user attributes with the network structural information. Eventually, the task to infer the missing value of the target attribute is switched to find the optimal feature weight vector. Please note that our work just employ the idea of Supervised Random Walk in [15], but they are totally different. Firstly, they solve different problems. Link prediction problem was solved in [15], while our work aims to solve the attribute inference problem. Secondly, different optimization objectives and optimization methods are designed for these two different problems.

3.5 The Optimization Function and Its Solution

Now, the question is how to estimate the weight parameters ω through an optimization objective. Note that the visiting probability p^T denotes the likelihood of user sharing the same target attribute value with seed users. So the "good result" is that the visiting probability of user u (denoted as p_u) should be relatively high if u is a positive example. A handful of objective functions expressing this need could be an option. Equation (7) gives one example. The impact of different choices of objective functions will be studied in the experiment section.

$$J(\omega) = \frac{\sum_{u \in V^L_{ext}} p_u}{\sum_{v \in V} p_v} \tag{7}$$

We use gradient ascent to maximize the objective function. Thus, we need to obtain the derivative of objective function with respect to ω as follow:

$$\frac{\partial J(\omega)}{\partial \omega} = \frac{\sum_{u \in V^L_{ext}} \frac{\partial p_u}{\partial \omega} \cdot \sum_{v \in V} p_v - \sum_{u \in V^L_{ext}} p_u \cdot \sum_{v \in V} \frac{\partial p_v}{\partial \omega}}{(\sum_{v \in V} p_v)^2} \tag{8}$$

p_u and p_v represent the visiting probability of user u and v respectively. $\frac{\partial p_u}{\partial \omega}$ and $\frac{\partial p_v}{\partial \omega}$ denote the derivative of p_u and p_v with respect to ω respectively. So we need to establish the connection between the parameters ω and the random walk scores p. The visiting probability p_u can be written as Eq. (9):

$$p_u = \sum_j p_j Q_{ju} \tag{9}$$

According to Eq. (9), we can deduce $\frac{\partial p_u}{\partial \omega}$ as follow:

$$\frac{\partial p_u}{\partial \omega} = \sum_j \frac{\partial p_j}{\partial \omega} Q_{ju} + \sum_j p_j \frac{\partial Q_{ju}}{\partial \omega} \tag{10}$$

If $(u, v) \in E$, $\frac{\partial Q_{ju}}{\partial \omega}$ can be written as Eq. (11):

$$\begin{aligned} \frac{\partial Q_{ju}}{\partial \omega} = & \frac{\frac{\partial f_\omega(\omega^T X_e(j,u))}{\partial \omega}(\sum_{i \in N(j)} f_\omega(\omega^T X_e(j, i)))}{(\sum_{i \in N(j)} f_\omega(\omega^T X_e(j, i)))^2} \\ & - \frac{f_\omega(\omega^T X_e(j, u))(\sum_{i \in N(j)} \frac{\partial f_\omega(\omega^T X_e(j,i))}{\partial \omega})}{(\sum_{i \in N(j)} f_\omega(\omega^T X_e(j, i)))^2} \end{aligned} \tag{11}$$

$X_e(j, u)$ and $X_e(j, i)$ denote the feature vector of edge (j, u) and edge (j, i) respectively. $f_\omega(\omega^T X_e(j, u))$ and $f_\omega(\omega^T X_e(j, i))$ represent the edge weight of edge (j, u) and edge (j, i) respectively. $N(j)$ denotes the neighbors of user j. So far we have described how to compute $\frac{\partial p}{\partial \omega}$. Then we use an iterative power-iterator like algorithm proposed by Backstrom et al. [15] to compute the random walk visiting probability vector p^T and its derivative $\frac{\partial p}{\partial \omega}$. And the pseudo code of gradient ascent is shown in Algorithm 2. While the

Algorithm 2 Gradient Ascent

Input: training instances, learning rate λ
Output: the optimal ω
for each $k = 1, \ldots, \|\omega\|$ **do**
  $\omega_k^{(0)} = 0$
end for
$t = 1$
while $J(\omega)$ has not converged **do**
  compute p^T and $\frac{\partial p}{\partial \omega}$ according to Alg.3
  for each $k = 1, \ldots, \|\omega\|$ **do**
    $\omega_k^{(t)} = \omega_k^{(t-1)} + \lambda \times \frac{\partial J(\omega)}{\partial \omega}$
  end for
  $t = t + 1$
end while

Algorithm 3 Derivatives of the Random Walk

Input: parameter ω
Output: the PageRank scores p and derivatives $\frac{\partial p}{\partial \omega}$
for each $u \in V$ **do**
$p_u^{(0)} = \frac{1}{|V|}$
end for
for each $u \in V$ and $k = 1, \ldots, \|\omega\|$ **do**
$\frac{\partial p_u^{(0)}}{\partial \omega_k} = 0$
end for
compute transition matrix Q with ω
$t = 1$
while p has not converged **do**
for each $u \in V$ **do**
$p_u^{(t)} = \sum_j p_j^{(t-1)} Q_{ju}$
end for
$t = t + 1$
end while
$t = 1$
for each $k = 1, \ldots, \|\omega\|$ **do**
while $\frac{\partial p}{\partial \omega}$ has not converged **do**
for each $u \in V$ **do**
compute $\frac{\partial p^{(t)}}{\partial \omega}$ according to (10)
end for
$t = t + 1$
end while
end for

iterative process to compute the derivatives of the random walk can be summarized by Algorithm 3.

4. Experiments

In this section we evaluate our method on two real datasets: Sina Weibo dataset and the Telecom dataset.

4.1 Datasets

Sina Weibo Dataset is crawled from the most popular Chinese social network platform Sina Weibo (http://weibo.com/). This dataset includes 34,199 users and 691,522 links among them. We also crawl profile information of these users, including user name, location, tags and 200 lasted tweets. We manually labeled these data and there are 5323 positive examples (whose affiliation is CMCC) as ground truth.

Telecom Dataset comes from the first big data contest in China (http://www.bigcloudsys.com/ccf2013/de-tail2.html). This dataset provides the call and message records of users. There is an edge between two users only if they have messaged each other. There are 54,487 users and 304,998 links in the network. We have known that 43,231 positive examples (provided by the big data contest committee) as ground truth. That is, the role of these users is student. Table 1 shows the basic statistics of these two datasets.

4.2 Baseline Methods

In this section, we compare the performances of SRW-COND with the following four representative methods.

SVM. We consider the attribute inference as a binary classification. We use weka's implementation of SVM in our experiments. The features for the classifier are derived from tweet messages and network structure. We have tried different kernels of SVM and we only show the results of SVM with a linear kernel because it wins over others.

Random Walk with Restart (RWR) [18]. This method propagates seed nodes' information to other nodes with fixed propagation probability.

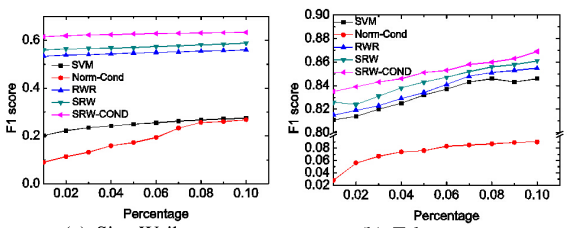
Supervised Random Walk (SRW) [15]. This method propagates seed nodes' information to other nodes with varying propagation probability. This method is simplified from SRW-COND by removing the seed extension phase for the purpose of a clear illustration of the impact of the first phase.

Norm-Cond. [5]. Norm-Cond. is a community detection like method for attribute inference, which greedily extends seed nodes with *Normalized Conductance*. This method serves as a benchmark for other methods that can be applied in the situation of small labeled data.

4.3 Experimental Results

4.3.1 Effectiveness Study of SRW-COND

These experiments validate effectiveness of SRW-COND through randomly selecting a very small proportion of positive examples (varying from 1% to 10%) as training set. The same restart probability 0.5 is used for SRW-COND, RWR, and SRW. Other parameters are set as suggested in the literals. The experiments are repeated 50 times and the F1 score is used to validate the performances of these models. Figure 2 shows the average results. The results show that SRW-COND always achieves best performances on all conditions. SVM simply combines structural and attribute information, so it has bad performance. Because of only considering the network structure, Norm-Cond. also performs badly. SRW



(a) Sina Weibo (b) Telecom

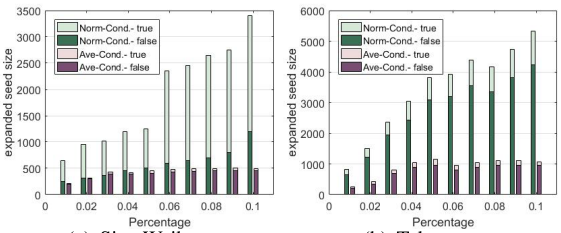
Fig. 2 Accuracy comparison of different algorithms.

Table 1 Statistics of datasets.

Datasets	Users	Edges	Average Degrees	Positive Examples
Sina Weibo Dataset	34,199	691,522	20.22	5323
Telecom Dataset	54,487	304,998	5.59	43,231

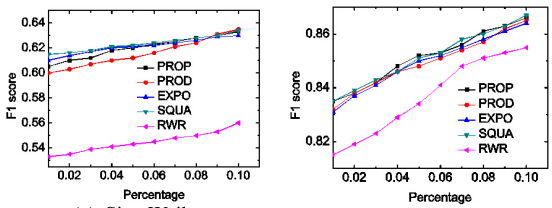
Table 2 Objective functions.

Proportional Function (PROP)	Production Function (PROD)	Exponential Function (EXPO)	SquareError Function (SQUA)
$\frac{\sum_{u \in V^L_{ext}} p_u}{\sum_{v \in V} p_v}$	$\prod_{u \in V^L_{ext}} p_u$	$\sum_{u \in V^L_{ext}} p_u^\beta$ and $\beta = -1$	$(p_v - p_u)^2$ if $p_u < p_v$ $u \in V^L_{ext}$, $v \in V^{L_N}$



(a) Sina Weibo (b) Telecom

Fig. 3 Accuracy comparison of labeled data extension.



(a) Sina Weibo (b) Telecom

Fig. 4 Effect of objective functions.

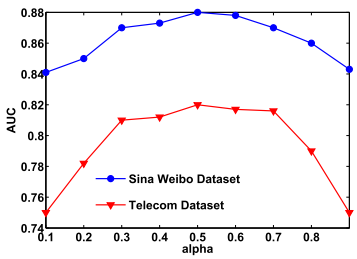


Fig. 5 Effect of restart probability alpha.

can get a good performance, but it is not so good as SRW-COND due to the lack of enough labeled data. Comparing the results on these two datasets, we can find that the performances of almost all methods on Telecom dataset are better than that on Sina Weibo dataset. We think the reason lies in that the task in Telecom is relatively easy. In other words, the task that infers the affiliation of a user in Sina Weibo dataset is more difficult than the task that infers whether a user is a student in Telecom dataset. We know that the roles of users in Telecom dataset are very limited, and the students have significant communication patterns. We can also find that, with the decrease of the percentage of labeled data, the superiority of SRW-COND tends to be more significant, which further show the benefits of SRW-COND on very small labeled data.

4.3.2 Effectiveness Study of Labeled Data Extension

Since the extension of labeled data is a critical step of SRW-COND, these experiments will validate its effectiveness of the extension through observing the number of right and wrong nodes generated by Ave-Cond. and Norm-Cond.. The experiments are repeated 50 times and the average results are shown in Fig. 3. We can see that Ave-Cond. obtains a small number of nodes with high accuracy, comparing with Norm-Cond. getting a large number of nodes with low accuracy. We believe the reasons lie in the following two aspects. Firstly, Ave-Cond. uses *Conductance* instead of *Normalized Conductance* as in Norm-Cond., which more effectively guarantees the quality of added nodes. Secondly, The stopping criterion of Ave-Cond. is stricter, which makes Ave-Cond. only extend a small number of nodes with high accuracy.

4.3.3 Impact of Optimization Objectives

These experiments illustrate the impact of optimization objectives. Same parameters are set as above experiments except the optimization objects. The objective functions are shown in Table 2. The results are shown in Fig. 4. We can see that different loss functions have subtle influence on the performances of our method on both datasets. The square error function slightly wins over other methods, because it incorporates the information of negative examples. Overall, SRW-COND with different functions significantly outperforms the baseline RWR.

4.3.4 Effect of Restart Probability

To illustrate how the different restart probabilities affect the performance of SRW-COND, we observe its performances with varying restart probabilities under the 10% of labeled data as training data. Figure 5 illustrates how AUC (Area under ROC Curve) change by differing α from 0.1 to 0.9. The results show that SRW-COND consistently has good performances when $\alpha \in [0.4, 0.7]$ on both datasets. Because in this case, the random walk process is able to make full use of labeled users as well as the global network structure. Finally, we set α to 0.5 in the experiments.

5. Conclusions

In this paper, we studied the attribute inference of users under very small labeled data and proposed the SRW-COND solution. The SRW-COND firstly employs a simple but effective greedy algorithm to extend seed set with high accuracy. Then it adapts a supervised random walk process

to make full use of network structure and attributes information of users. Experiments on Sina Weibo dataset and Telecom dataset demonstrate that, under small training set, SRW-COND can significantly outperform other well established methods.

References

[1] G. Zeng, P. Luo, E. Chen, and M. Wang, "From Social User Activities to People Affiliation," In: ICDM, pp.1277–1282, 2013.
[2] N.K. Sharma, S. Ghosh, F. Benevenuto, N. Ganguly, and K. Gummadi, "Inferring Who-is-Who in the Twitter Social Network," ACM SIGCOMM Computer Communication Review, vol.42, no.4, pp.533–538, 2012.
[3] F. Zamal, A. Liu, and W. Ruths, "Homophily and Latent Attribute Inference: Inferring Latent Attributes of Twitter Users from Neighbors," In: ICWSM, pp.387–390, 2012.
[4] N.Z. Gong, A. Talwalkar, L. Mackey, L. Huang, E.C.R. Shin, E. Stefanov, E. Shi, and D. Song, "Joint Link Prediction and Attribute Inference Using a Social-Attribute Network," ACM TIST, vol.5, no.2, pp.1–20, 2014.
[5] A. Mislove, B. Viswanath, K.P. Gummadi, and P. Druschel, "You Are Who You Know: Inferring User Profiles in Online Social Networks," In: WSDM, pp.251–260, 2010.
[6] Y. Ding, S. Yan, Y.B. Zhang, W. Dai, and L. Dong, "Predicting the attributes of social network users using a graph-based machine learning method," Computer Communications, vol.73, pp.3–11, 2015.
[7] E.M. Ardehaly and A. Culotta, "Inferring latent attributes of Twitter users with label regularization," In: HLT-NAACL, pp.185–195, 2015.
[8] T.L. Fond and J. Neville, "Randomization tests for distinguishing social influence and homophily effects," In: WWW, pp.601–610, 2011.
[9] L. Backstrom, E. Sun, and C. Marlow, "Find me if you can: improving geographical prediction with social and spatial proximity," In: WWW, pp.61–70, 2010.
[10] D. Rao, D. Yarowsky, A. Shreevats, and M. Gupta, "Classifying Latent User Attributes in Twitter," In: SMUC, pp.37–44, 2010.
[11] M. Pennacchiotti and A.M. Popescu, "A Machine Learning Approach to Twitter User Classification," In: ICWSM, pp.281–288, 2011.
[12] X. Yan and L. Yan, "Gender Classification of Weblog Authors," AAAI Spring Symposia on Computational Approaches, pp.228–230, 2006.
[13] M. Roth, A. Ben-David, D. Deutscher, G. Flysher, I. Horn, A. Leichtberg, N. Leiser, Y. Matias, and R. Merom, "Suggesting Friends Using the Implicit Social Graph," In: KDD, pp.233–242, 2010.
[14] E. Zheleva and L. Getoor, "To Join or Not to Join: The Illusion of Privacy in Social Networks with Mixed Public and Private User Profiles," In: WWW, pp.531–540, 2009.
[15] L. Backstrom and J. Leskovec, "Supervised Random Walks: Predicting and Recommending Links in Social Networks," In: WSDM, pp.635–644, 2011.
[16] J. Leskovec, K.J. Lang, A. Dasgupta, and M.W. Mahoney, "Statistical Properties of Community Structure in Large Social and Information Networks," In: WWW, pp.695–704, 2008.
[17] M. McPherson, L. Smith-Lovin, and J.M. Cook, "Birds of a Feather: Homophily in Social Networks," In: Annual Review of Sociology, vol.27, no.1, pp.415–444, 2001.
[18] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast Random Walk with Restart and Its Applications," In: ICDM, pp.613–622, 2006.

Ding Xiao received the B.S. degree from the Paris Dauphine University in 1991, and the M.S. degree from UPMC France in 1993. Joined the School of Computer Science of BUPT China in 1995.

Rui Wang received the B.S. degree from the Beijing University of Posts and Telecommunications in 2014. She is currently a master student in BUPT. Her research interests are in machine learning and data mining.

Lingling Wu received the B.S. degree from Beijing Information Science and Technology University in 2012, the M.S. degree from Beijing University of Posts and Telecommunications in 2015. She is working as a software developer in Baidu Inc. since 2015 and her research interests are in machine learning, data mining.