2013 IEEE 13th International Conference on Data Mining

From Social User Activities to People Affiliation

Guangxiang Zeng[1], Ping Luo[2], Enhong Chen[1] and Min Wang[3]
[1]University of Science and Technology of China, [2]HP Labs China, [3]Google Research
[1]{zgx@mail., cheneh@}ustc.edu.cn, [2]ping.luo@hp.com, [3]minwang@google.com

***Abstract*—This study addresses the problem of inferring users' employment affiliation information from social activities. It is motivated by the applications which need to monitoring and analyzing the social activities of the employees from a given company, especially their social tracks related to the work and business. It definitely helps to better understand their needs and opinions towards certain business area, so that the account sales targeting these customers in the given company can adjust the sales strategies accordingly.**

Specifically, in this task we are given a snapshot of a social network and some labeled social users who are the employees of a given company. Our goal is to identify more users from the same company. We formulate this problem as a task of classifying nodes over a graph, and develop a *Supervised Label Propagation* model. It naturally incorporates the rich set of features for social activities, models the networking effect by label propagation, and learns the feature weights so that the labels are propagated to the right users. To validate its effectiveness, we show our case studies on identifying the employees of "China Telecom" and "China Unicom" from *Sina Weibo*. The experimental results show that our method significantly outperforms the compared baseline ones.

I. Introduction

With the proliferation of social media and portable devices, people spend more and more time on social media platforms, which attracts increasing studies on mining business insights and actionable knowledge from social media. However, most of these previous works only consider social users as *mass consumers*, and focus on their tracks in using *consumer products*. For example, [1] monitors the experiences and sentiments of social users on consumer products, and [2], [3] identify the trustful and influential users for consumer product promotion.

Besides mass consumers, social users are often acting as *enterprise employees*. As social media is penetrating into their everyday work, there has occurred a shift on how enterprise employees receive the business information. As shown in Figure 1, in the past sales people were the main information source of enterprise buyers. However, nowadays enterprise customers turn to social media for business and technical information much more frequently. Therefore, to succeed in enterprise business we need to monitor what the enterprise customers read, comment and retweet about their business topics in social media. By summarizing these customer tracks in social media, the sales team can better plan the sales goals and strategies.

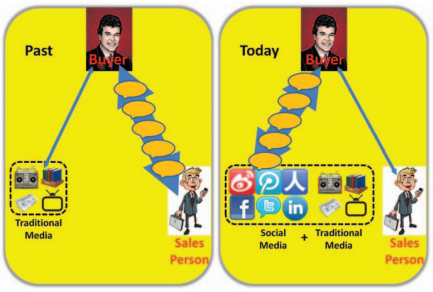


Figure 1. The Information Sources of Buyers.

To this end, the first task is to identify the employees of a given company in social media. Thus, in this study we focus on this problem of profiling users' employment affiliations in the context of social networks. Given a snapshot of a social network and some labeled social users who (we already know) are either employees or non-employees of a given company[1], our goal is to identify more social users from the same company. Temporal change of user affiliation is out of the scope of this study.

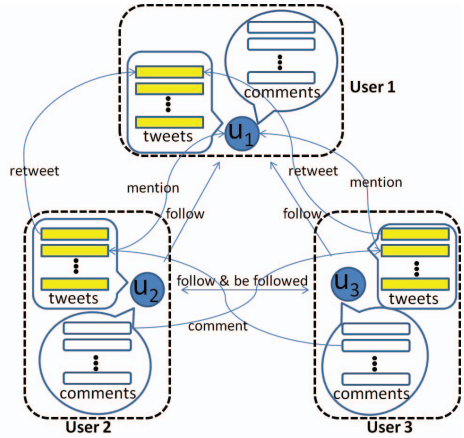


Figure 2. An Snapshot Example of Sina Weibo Social Networks.

Clearly, this problem can be formulated as a supervised classification task for the nodes over a graph. However, it is challenging in the following aspects. First, as shown in Figure 2, the features we can use for classification are based on the various types of social activities, including following users, posting tweets, retweeting, commenting

[1]There are only a small amount of users who provide their affiliation information when registration.

1550-4786/13 $31.00 © 2013 IEEE
DOI 10.1109/ICDM.2013.101

tweets[2], mentioning users in tweets, etc. We need a uniform model, which can not only incorporate all these features but also infer the contribution of each feature for classification automatically. Second, we need to consider the networking effect in social media for this task. Usually, the colleagues from a company may follow each other, and discuss the content related to their company in social media. Learning the feature weights in the context of networks makes this task more complicated.

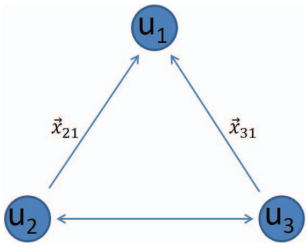


Figure 3. The Homogeneous Graph of Users.

Since we only focus on the classification of users in this study, we can simplify this heterogeneous graph to a homogeneous graph, as shown in Figure 3, where only the nodes of users remain and the interactive activities between two users are summarized as the features on the edge between them.

With this simple representation of social graph, we propose a framework of *Supervised Label Propagation*, SLP for short. In this framework, the label of each node is the weighted sum of its initial label information and the labels of its neighbors. The degree that how much the label of a node is affected by its neighbors is the function of their corresponding features (depicting the social activities between each pair of neighbors) and the feature weights. These weights can be learned in a proposed optimization problem so that the class labels are more likely to propagate onto the users working for the given company. We summarize the contributions of this study as follows.

- To the best knowledge of ours, we are the first to infer users' affiliation on social networks, and formulate this problem as a task of node classification over graph.
- We propose a supervised label propagation framework to address this problem.
- To demonstrate the effectiveness of the proposed model we use the data crawled from Sina Weibo and focus on identifying the users from the two biggest Chinese telecommunication companies, namely "China Unicom" and "China Telecom". The results show the significant improvement of the proposed method over the compared baseline methods.

The remainder of this paper is organized as follows. We formulate the problem of *User Affiliation Inferring* on social networks in Section II. Then, the label propagation process is introduced in Section III. We present the *Supervised Label Propagation* model and the model learning method in Section IV. Section V discusses the features we use to summarize the social activities. Experimental studies are presented in Section VI. Finally, we give out the related work and conclude this paper in Section VII.

[2]In Chinese social media a comment to a tweet is only attached to the targeted tweet, but does not appear in the timeline of the comment's author. Thus, we distinguish comments with tweets in this study.

II. Problem Statement

Notions & Denotations. All the notions and denotations used in this study are summarized in Table I. With these symbols we can derive more interesting concepts. For example, $\mathcal{T}(u_j) \cap \mathcal{R}(u_i)$ is the set of tweets which are posted by u_j and retweeted by u_i, and $\mathcal{T}(u_j) \cap \mathcal{C}(u_i)$ is the set of tweets which are posted by u_j and commented by u_i. $\mathcal{T}(u_i) - \widetilde{\mathcal{R}}(u_i)$ is the set of tweets originally created by u_i.

Table I
List of Symbols

$\mathcal{U}$	The set of users.
$\mathcal{E}$	The set of edges.
G	A *Sina Weibo* social network, $G = (\mathcal{U}, \mathcal{E})$.
u	A social media user.
i, j	Index over users.
$\langle u_i, u_j \rangle$	A directed edge means user u_i follow user u_j.
m	A tweet.
$\mathcal{I}(u_i)$	The followers of u_i.
$\mathcal{O}(u_i)$	The followees of u_i.
$\mathcal{N}(u_i)$	The neighbors of u_i in social networks, includes both followers and followees. $\mathcal{N}(i)$ for short.
$\mathcal{T}(u_i)$	Tweets posted by u_i.
$\mathcal{R}(u_i)$	Tweets retweeted by u_i.
$\widetilde{\mathcal{R}}(u_i)$	Tweets retweeted by u_i and whose original authors are not u_i.
$\mathcal{C}(u_i)$	Tweets to which u_i comments.
$\mathcal{M}(u_i)$	Tweets which mention u_i.
$\mathcal{U}_P^L$	Set of labeled positive users.
$\mathcal{U}_N^L$	Set of labeled negative users.
$\mathcal{T}_P^L$	Set of labeled positive tweets.
$\mathcal{T}_N^L$	Set of labeled negative tweets.
P_i	Likelihood of u_i belonging to the affiliation Ω.
P_i^0	Label bias of u_i.
σ_i	Influence of u_i.
r_{ji}	Edge association over the edge from u_j and u_i.
ω_{ji}	Label propagation strength from u_j to u_i, $\omega_{ji} = r_{ji} \cdot \sigma_j$.
W	Matrix of label propagation strength , $W = [\omega_{ij}]$.

Inferring User Affiliation. This study aims at discovering the employees of a given company Ω in a social network. Here, we are given a social network $G = (\mathcal{U}, \mathcal{E})$, the tweets and comments posted by each user $u_i \in \mathcal{U}$. Also, we have a small set of labeled users $\mathcal{U}^L = \mathcal{U}_P^L \cup \mathcal{U}_N^L$, where $\mathcal{U}_P^L$ are the employees of Ω and $\mathcal{U}_N^L$ are not the employees of company Ω. This small set of labels is provided by the users who filled in their affiliation information when registration. With these data, we aim to find more employees of Ω besides those in $\mathcal{U}_P^L$.

In this task we will assign a real likelihood score P_i to measure how likely that a user u_i is an employee of Ω. The bigger the likelihood score P_i is, the more likely that the user u_i belongs to Ω. Next, we will propose a model to calculate it.

III. Label Propagation Process

A. Label Propagation

Motivated by the observation that users with common attributes are more likely to be friends [4], the framework of label propagation [5] is adopted in this study. In this framework, the label of a node is affected by not only its local label information but also the labels of its neighbors.

$$P_i^{[t+1]} = (1-\eta) \sum_{u_j \in \mathcal{N}(u_i)} \omega_{ji} \cdot P_j^{[t]} + \eta P_i^0 \tag{1}$$

where $0 < \eta < 1$, ω_{ji} is label propagation strength from u_j to u_i, η is the restart probability of label information that P_i jumps to its personal label bias P_i^0. The updates continue until convergence. Let $W = [\omega_{ij}]$, then the close form of $\vec{P}$:

$$\vec{P} = \eta[I - (1-\eta)W^T]^{-1}\vec{P}^0. \tag{2}$$

B. Label Propagation Strength ω_{ji}

The label propagation strength ω_{ji} is jointly affected by two factors, namely the influence σ_j of u_j and the edge association r_{ji} from u_j to u_i.

$$\omega_{ji} = \sigma_j \cdot r_{ji} \tag{3}$$

1) Edge Association r_{ji} : According to [6], it can be represented as follows,

$$r_{ji} = \frac{f(\vec{\alpha} \cdot \vec{x}_{ji})}{\sum_{l \in \mathcal{N}(i)} f(\vec{\alpha} \cdot \vec{x}_{li})}, \tag{4}$$

where $\vec{x}_{ji}$ depicts the features on the interaction from u_j and u_i, which is detailed in Section V-A. $\vec{\alpha}$ is the weight vector on these features, and $f(x) = \frac{1}{1+e^{-x}}$.

2) Influence σ_j : Here, we try to model the influence on attracting the working colleagues as her social friends. Specifically, for the influence of node j we have

$$\sigma_j = g(\vec{\beta} \cdot \vec{y}_j) \tag{5}$$

where $\vec{\beta}$ is the weight vector on the feature vector $\vec{y}_j$, and the function $g = \frac{1}{1+e^{-x}}$ is adopted here. Section V-B will detail the features of $\vec{y}_j$.

C. Label Bias P_i^0

$$P_i^0 = h(\vec{\gamma} \cdot \vec{z}_i), \tag{6}$$

where $\vec{z}_i$ denotes this set of features for label bias, which is detailed in Section V-C. $\vec{\gamma}$ is the weight vector on these features, and $h(x) = 1 - e^{-x}$.

As we can see from the above, the final likelihood vector $\vec{P}$ is determined by the parameter vector $\vec{\theta} = [\vec{\alpha}, \vec{\beta}, \vec{\gamma}]$. Next, we will show you how these parameters are learned based on the labeled data.

IV. Supervised Label Propagation

A. Model Formulation

In order to infer parameters $\vec{\theta} = [\vec{\alpha}, \vec{\beta}, \vec{\gamma}]$, we formulate the following optimization problem with the labeled data

$$\min J(\vec{\theta}) = \sum_{i \in \mathcal{U}_P^L} \sum_{j \in \mathcal{U}_N^L} S(P_j - P_i) + C \cdot ||\vec{\theta}||^2, \tag{7}$$

where $S(x) = \frac{1}{1+e^{-\mu x}}$, μ is empirically set to 500.0. C is regularization parameter to control the model complexity[3].

The intuition of Equation (7) is that we identity the parameters such that the likelihood of the positive users is larger than that of negative users as much as possible. We use the gradient descent method to solve Equation (7). The main challenge is how to compute $\frac{\partial J(\vec{\theta})}{\partial \vec{\theta}} = [\frac{\partial J(\vec{\theta})}{\partial \vec{\alpha}}, \frac{\partial J(\vec{\theta})}{\partial \vec{\beta}}, \frac{\partial J(\vec{\theta})}{\partial \vec{\gamma}}]$.

B. Model Learning

Let $\delta_{ji} = P_j - P_i$, α_k denotes the k-th entity of $\vec{\alpha}$, β_p denotes the p-th entity of $\vec{\beta}$ and γ_q denotes the q-th entity of $\vec{\gamma}$. Then we have:

$$\begin{aligned} \frac{\partial J(\vec{\theta})}{\partial \alpha_k} &= \sum_{i \in \mathcal{U}_P^L} \sum_{j \in \mathcal{U}_N^L} \frac{\partial S(P_j - P_i)}{\partial \alpha_k} + 2C \cdot \alpha_k \\ &= \sum_{i \in \mathcal{U}_P^L} \sum_{j \in \mathcal{U}_N^L} \frac{\partial S(\delta_{ji})}{\partial \delta_{ji}} (\frac{\partial P_j}{\partial \alpha_k} - \frac{\partial P_i}{\partial \alpha_k}) + 2C \cdot \alpha_k \end{aligned} \tag{8}$$

$$\begin{aligned} \frac{\partial J(\vec{\theta})}{\partial \beta_p} &= \sum_{i \in \mathcal{U}_P^L} \sum_{j \in \mathcal{U}_N^L} \frac{\partial S(P_j - P_i)}{\partial \beta_p} + 2C \cdot \beta_p \\ &= \sum_{i \in \mathcal{U}_P^L} \sum_{j \in \mathcal{U}_N^L} \frac{\partial S(\delta_{ji})}{\partial \delta_{ji}} (\frac{\partial P_j}{\partial \beta_p} - \frac{\partial P_i}{\partial \beta_p}) + 2C \cdot \beta_p \end{aligned} \tag{9}$$

$$\begin{aligned} \frac{\partial J(\vec{\theta})}{\partial \gamma_q} &= \sum_{i \in \mathcal{U}_P^L} \sum_{j \in \mathcal{U}_N^L} \frac{\partial S(P_j - P_i)}{\partial \gamma_q} + 2C \cdot \gamma_q \\ &= \sum_{i \in \mathcal{U}_P^L} \sum_{j \in \mathcal{U}_N^L} \frac{\partial S(\delta_{ji})}{\partial \delta_{ji}} (\frac{\partial P_j}{\partial \gamma_q} - \frac{\partial P_i}{\partial \gamma_q}) + 2C \cdot \gamma_q \end{aligned} \tag{10}$$

$\frac{\partial S(\delta_{ji})}{\partial \delta_{ji}}$ is obvious, but $\frac{\partial P_i}{\partial \alpha_k}$, $\frac{\partial P_i}{\partial \beta_p}$ and $\frac{\partial P_i}{\partial \gamma_q}$ is not so clear. Let x_{ji}^k denotes the k-th entity of $\vec{x}_{ji}$, y_i^p denotes the p-th entity of $\vec{y}_i$ and z_i^q denotes the q-th entity of $\vec{z}_i$. According to the update rule in Equation (1), we have:

$$\frac{\partial P_i}{\partial \alpha_k} = (1-\eta) \sum_{j \in \mathcal{N}(i)} [\frac{\partial \omega_{ji}}{\partial \alpha_k} P_j + \omega_{ji} \frac{\partial P_j}{\partial \alpha_k}] \tag{11}$$

$$\frac{\partial P_i}{\partial \beta_p} = (1-\eta) \sum_{j \in \mathcal{N}(i)} [\frac{\partial \omega_{ji}}{\partial \beta_p} P_j + \omega_{ji} \frac{\partial P_j}{\partial \beta_p}] \tag{12}$$

$$\frac{\partial Pi}{\partial \gamma_q} = (1-\eta) \sum_{j \in \mathcal{N}(i)} \omega_{ji} \frac{\partial P_j}{\partial \gamma_q} + \eta \frac{\partial P_i^0}{\partial \gamma_q} \tag{13}$$

In the above three equations we have

$$\begin{aligned} &\frac{\partial \omega_{ji}}{\partial \alpha_k} = \sigma_j \cdot \frac{\partial r_{ji}}{\partial \alpha_k} \\ &\sigma_j \cdot \frac{x_{ji}^k f'(\vec{\alpha} \cdot \vec{x}_{ji}) \sum_{l \in \mathcal{N}(i)} f(\vec{\alpha} \cdot \vec{x}_{li}) - f(\vec{\alpha} \cdot \vec{x}_{ji}) \sum_{l \in \mathcal{N}(i)} x_{li}^k f'(\vec{\alpha} \cdot \vec{x}_{li})}{[\sum_{l \in \mathcal{N}(i)} f(\vec{\alpha} \cdot \vec{x}_{li})]^2} \end{aligned} \tag{14}$$

[3]In this study we set $C = 0.1$ empirically since we found that the proposed methods are not sensitive to C.

$$\frac{\partial \omega_{ji}}{\partial \beta_p} = r_{ji} \cdot \frac{\partial \sigma_j}{\partial \beta_p} = r_{ji} \cdot y_j^p \cdot g'(\vec{\beta} \cdot \vec{y}_j) \tag{15}$$

$$\frac{\partial P_i^0}{\partial \gamma_q} = z_i^q \cdot h'(\vec{\gamma} \cdot \vec{z}_i) \tag{16}$$

With Equations (11) through (16) we can compute the derivatives $\frac{\partial P_i}{\partial \alpha_k}$, $\frac{\partial P_i}{\partial \beta_p}$ and $\frac{\partial Pi}{\partial \gamma_q}$ in an iterative process, as shown in Algorithm 1. The convergence of this algorithm is similar to those of the power-iteration [7]. After we get $\frac{\partial \vec{P}}{\partial \vec{\theta}}$, we can compute the gradient of $J(\vec{\theta})$ by Equations (8) through (10). Then, we can apply the gradient descent method directly to minimize $J(\vec{\theta})$.

Algorithm 1 Compute Gradient

Input: $\mathcal{U} = \{u_i\}$, $W = [\omega_{ij}]$, $\vec{P}$, $\vec{P}^0$;
Output: $\frac{\partial \vec{P}}{\partial \vec{\theta}} = [\frac{\partial \vec{P}}{\partial \vec{\alpha}}, \frac{\partial \vec{P}}{\partial \vec{\beta}}, \frac{\partial \vec{P}}{\partial \vec{\gamma}}]$;

initialize $\frac{\partial P_i}{\partial \vec{\alpha}}^{[0]}$, $\frac{\partial P_i}{\partial \vec{\beta}}^{[0]}$ and $\frac{\partial P_i}{\partial \vec{\gamma}}^{[0]}$, $t = 0$;
while $\frac{\partial \vec{P}}{\partial \vec{\theta}}$ not converged **do**
  for each u_i, k, p, q **do**
    $\frac{\partial P_i}{\partial \alpha_k}^{[t+1]} = (1-\eta) \sum_{j \in \mathcal{N}(i)} [\frac{\partial \omega_{ji}}{\partial \alpha_k} P_j + \omega_{ji} \frac{\partial P_j}{\partial \alpha_k}^{[t]}]$;
    $\frac{\partial P_i}{\partial \beta_p}^{[t+1]} = (1-\eta) \sum_{j \in \mathcal{N}(i)} [\frac{\partial \omega_{ji}}{\partial \beta_p} P_j + \omega_{ji} \frac{\partial P_j}{\partial \beta_p}^{[t]}]$;
    $\frac{\partial Pi}{\partial \gamma_q}^{[t+1]} = (1-\eta) \sum_{j \in \mathcal{N}(i)} \omega_{ji} \frac{\partial P_j}{\partial \gamma_q}^{[t]} + \eta \frac{\partial P_i^0}{\partial \gamma_q}$;
  end for
  $t = t + 1$;
end while
return $\frac{\partial \vec{P}}{\partial \vec{\theta}} = [\frac{\partial \vec{P}}{\partial \vec{\alpha}}, \frac{\partial \vec{P}}{\partial \vec{\beta}}, \frac{\partial \vec{P}}{\partial \vec{\gamma}}]$;

C. Method Summary

Given all the features detailed in Section V, each iteration of the learning process contains the two steps: 1) given the current parameters settings $\{\vec{\alpha}, \vec{\beta}, \vec{\gamma}\}$, compute $\vec{P}$ iteratively by the label propagation in Equation (1). 2) with the current $\vec{P}$, optimize the parameters of $\{\vec{\alpha}, \vec{\beta}, \vec{\gamma}\}$ by the gradient descent method, detailed in Section IV-B. These two steps are conducted iteratively until $\vec{P}$ becomes unchange. In our experiments $\vec{P}$ becomes stable in about 200 iterations.

V. Features

A. Features $\vec{x}_{ji}$ in Edge Association r_{ji}

We use social activities among users to portray the edge association. There are four kinds of social activities in this study: 1) *follow* activities; 2) *retweet* activities; 3) *comment* activities; 4) *mention* activities. Beside social activities, the content of interaction messages between users is important for identify the affiliation of users. So extract the topic information of the interaction messages between users.

To extract topics from interaction messages, we label two sets of tweets manually: $\mathcal{T}_P^L$ denotes the tweets related to the given company (belong to topic T_1), and $\mathcal{T}_N^L$ denotes the tweets irrelevant to the company (belong to topic T_2). Then we adopt supervised PLSA to estimate the probability values of $P(T_1|t)$ and $P(T_2|t)$ for any tweet t. Next, for any tweet set $\mathcal{T}$, we give a score to measure how much content in this set is related to the given company:

$$H(\mathcal{T}) = \sum_{t \in \mathcal{T}} P(T_1|t). \tag{17}$$

The feature of r_{ji} is separated into two parts, $\vec{x}_{ji} = [\vec{a}_{ji}, \vec{b}_{ji}]$. $\vec{a}_{ji}$ considers the association if u_j is the followee of u_i, and $\vec{a}_{ji} = \vec{0}$ if u_j is *not* the followee of u_i. Meanwhile, $\vec{b}_{ji}$ considers the association if u_j is the follower of u_i, and $\vec{b}_{ji} = \vec{0}$ if u_j is the *not* follower of u_i. It is not difficult to understand the follow equations with the symbols summarized in Table I.

1) The computing of $\vec{a}_{ji}$ when u_j is the followee of $\vec{u}_i$: We have the following 4 features. a_{ji}^l denotes the l-th entry in $\vec{a}_{ji}$, where $l = 1, 2, 3, 4$.

- The *follow* activity: $a_{ji}^1 = \frac{1}{||\mathcal{O}(u_i)||}$.
- The *retweet* activity: $a_{ji}^2 = \frac{H(\mathcal{R}(u_i) \cap \mathcal{T}(u_j))}{\sum_{u_e \in \mathcal{O}(u_i)} H(\mathcal{R}(u_i) \cap \mathcal{T}(u_e))}$.
- The *comment* activity: $a_{ji}^3 = \frac{H(\mathcal{C}(u_i) \cap \mathcal{T}(u_j))}{\sum_{u_e \in \mathcal{O}(u_i)} H(\mathcal{C}(u_i) \cap \mathcal{T}(u_e))}$.
- The *mention* activity:
$$a_{ji}^4 = \frac{H(\mathcal{T}(u_i) \cap \mathcal{M}(u_j) \cup \mathcal{T}(u_j) \cap \mathcal{M}(u_i))}{\sum_{u_e \in \mathcal{I}(u_i)} H(\mathcal{T}(u_i) \cap \mathcal{M}(u_e) \cup \mathcal{T}(u_e) \cap \mathcal{M}(u_i))}.$$

2) The computing of $\vec{b}_{ji}$ when u_j is the follower of u_i: We also have the following 4 features. b_{ji}^l denotes the l-th entry in $\vec{b}_{ji}$, where $l = 1, 2, 3, 4$.

- The *follow* activity: $b_{ji}^1 = \frac{1}{||\mathcal{I}(u_i)||}$.
- The *retweet* activity: $b_{ji}^2 = \frac{H(\mathcal{R}(u_j) \cap \mathcal{T}(u_i))}{\sum_{u_e \in \mathcal{I}(u_i)} H(\mathcal{R}(u_e) \cap \mathcal{T}(u_i))}$.
- The *comment* activity: $b_{ji}^3 = \frac{H(\mathcal{C}(u_j) \cap \mathcal{T}(u_i))}{\sum_{u_e \in \mathcal{I}(u_i)} H(\mathcal{C}(u_e) \cap \mathcal{T}(u_i))}$.
- The *mention* activity:
$$b_{ji}^4 = \frac{H(\mathcal{T}(u_i) \cap \mathcal{M}(u_j) \cup \mathcal{T}(u_j) \cap \mathcal{M}(u_i))}{\sum_{u_e \in \mathcal{I}(u_i)} H(\mathcal{T}(u_i) \cap \mathcal{M}(u_e) \cup \mathcal{T}(u_e) \cap \mathcal{M}(u_i))}$$

B. Features $\vec{y}_i$ in Influence σ_i

Features in $\vec{y}_i$ aim to depict the influence of u_i in attracting the working colleagues. We develop the three features. To keep the feature value in the range of [0, 1], we transform the original values by function $F(x) = 1 - \frac{1}{x+1}$. Similarly, y_i^l denotes the l-th entry in $\vec{y}_i$, where $p = 1, 2, 3$.

- Number of followers: $y_i^1 = F(\#\{\text{followers of } u_i\})$.
- Weighted sum of u_i's tweets, which are retweeted.
$$y_i^2 = F(\sum_{t \in \mathcal{T}(u_i)} P(T_1|t) \cdot N_{rt}(t)),$$
where $N_{rt}(t)$ is the number of times that tweet t has been retweeted.
- Weighted sum of u_i's tweets, which are commented.
$$y_i^3 = F(\sum_{t \in \mathcal{T}(u_i)} P(T_1|t) \cdot N_{cm}(t)),$$
where $N_{cm}(t)$ is the number of times that tweet t has been commented.

C. Features $\vec{z}_i$ in Label Bias P_i^0

To feature P_i^0, we not only make use the users' activities on content about the given company, but also their emotion to the given company. To detect emotion of a tweet, we leverage *emoticons* used tweets. We divide the emoticons into two groups, E_P denotes the positive emoticons and E_N denotes the negative emoticons. We design 5 features on label bias, z_i^l denotes the l-th entry of $\vec{z}_i$, $l = 1, 2, 3, 4, 5$.

- Tweets originally posted by u_i: $z_i^1 = \frac{H(\mathcal{T}(u_i) - \widetilde{\mathcal{R}}(u_i))}{||\mathcal{T}(u_i) - \widetilde{\mathcal{R}}(u_i)||}$.
- Tweets retweeted by u_i: $z_i^2 = \frac{H(\mathcal{R}(u_i))}{||\mathcal{R}(u_i)||}$.
- Tweets commented by u_i: $z_i^3 = \frac{H(\mathcal{C}(u_i))}{||\mathcal{C}(u_i)||}$.
- Tweets posted by u_i with positive emotion:

$$z_i^4 = \frac{\sum_{m \in \mathcal{T}(u_i)} P(T_1|m) \cdot ||E(m) \cap E_P||}{||\mathcal{T}(u_i)||}.$$

- Tweets posted by u_i with negative emotion:

$$z_i^5 = \frac{\sum_{m \in \mathcal{T}(u_i)} P(T_1|m) \cdot ||E(m) \cap E_N||}{||\mathcal{T}(u_i)||}.$$

VI. Experimental Evaluation

A. Data Sets

Our two data sets are crawled from the most popular Chinese social network platform *Sina Weibo*. In this study we focus on the users from two Chinese telecommunication companies, namely "*China Telecom*" and "*China Unicom*" (*telecom* and *unicom* for short respectively). Table II shows the basic statistics of our two data sets. We denote *PT* as the set of users who are *verified* as employees of the company, and *NT* as the set of verified users who are not employees of the given companies.

Table II
Statistics of Data Sets

data set	#user	#edge	#tweet	#comment	#PT	#NT
China Telecom (*telecom*)	14,477	1,201,766	12,254,684	10,001,546	1,909	1,550
China Unicom (*unicom*)	7,187	374,674	7,835,021	6,564,512	710	1,420

B. Evaluation Methodology

We select $1/T$ ($T = 10$) of the labeled users ($1/T$ of PT and $1/T$ of NT, denoted as $\mathcal{U}_P^L$ and $\mathcal{U}_N^L$) for training. Then, the rest $(T-1)/T$ labeled users ($\{PT - \mathcal{U}_P^L\} \cup \{NT - \mathcal{U}_N^L\}$) are used for testing. We conduct 10-fold cross validation and report the average results. η is set to 0.1 for SLP methods.

PRate@pct, NRate@pct. We rank the users in the decrease order by their predicted likelihood scores. Ideally, the users in $\{PT - \mathcal{U}_P^L\}$ should be ranked above the users in $\{NT - \mathcal{U}_N^L\}$. Thus, after identifying the top $pct\%$ of the users, denoted by $\mathcal{U}_{pct}$, we calculate $PRate@pct$ and $NRate@pct$. Given a pct, the bigger the $PRate@pct$, the better the result. In contract, the smaller the $NRate@pct$, the better the result.

$$PRate@pct = \frac{||\mathcal{U}_{pct} \cap \{PT - \mathcal{U}_P^L\}||}{||\{PT - \mathcal{U}_P^L\}||} \times 100\%$$

$$NRate@pct = \frac{||\mathcal{U}_{pct} \cap \{NT - \mathcal{U}_N^L\}||}{||\{NT - \mathcal{U}_N^L\}||} \times 100\%$$

AUC [8]. Given $\mathcal{U}^+$ and $\mathcal{U}^-$ are the labeled positive set and the negative set. Our AUC measure is defined as:

$$AUC(\mathcal{U}^+, \mathcal{U}^-) = \frac{1}{||\mathcal{U}^+|| \cdot ||\mathcal{U}^-||} \sum_{i \in \mathcal{U}^+} \sum_{j \in \mathcal{U}^-} \Pi(P_i - P_j). \quad (18)$$

Thus, the AUC of training is $AUC(\mathcal{U}_P^L, \mathcal{U}_N^L)$, and the AUC of testing is $AUC(\{PT - \mathcal{U}_P^L\}, \{NT - \mathcal{U}_N^L\})$.

C. Baseline Methods

SVM. We use LIBSVM [9] with RBF kernel as our SVM baseline, and use grid search for getting best parameter settings. SVM_{node} and $SVM_{node+edge}$ represent the methods with node features only and all the features, respectively.

SLP (without influence). Remove the *Influence* variable from the SLP model.

SLP (fix label bias). Pre-train the weights $\vec{\gamma}$ with features of *Label Bias*. Then fix it in the SLP learning process.

D. Experimental Result

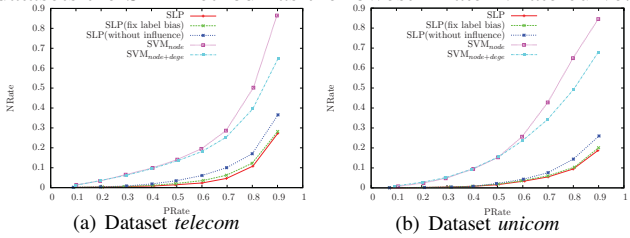
Table III
AUC Comparison.

Learning Method	Dataset *telecom*		Dataset *unicom*	
	TrainAUC	TestAUC	TrainAUC	TestAUC
SLP	0.924059977	**0.920894921**	0.939314195	**0.934514904**
$SLP(fixlabelbias)$	0.917705642	0.915913631	0.935846346	0.932365089
$SLP(withoutinfluence)$	0.891770619	0.888195797	0.923021509	0.918269617
SVM_{node}	0.970846439	0.724359834	0.981655709	0.701526671
$SVM_{node+edge}$	0.99950764	0.775220596	1.0	0.749070532

Comparison of AUC. From Table III we can see that the three SLP methods perform much better than the two SVM methods on testing data while their performance is not as good as that of SVM baselines on training data. This indicates that the generalization ability of the SLP methods is much better than that of SVM baselines in our problem.

By comparing SLP and SLP without influence, we can see that the introduction of *Influence* variable improves the AUC values by 3.68% on the dataset *telecom* and 1.77% on the dataset *unicom*. Lastly, the comparison between SLP and SLP with fixed label bias tells us that the co-training of all the model parameters produces better results.

Comparison of PRate & NRate on the Test Data. We use PRate as the x-axis, NRate as the y-axis, and plot the *PRate-NRate* curve of each method on the two datasets. For the same value of PRate, the lower PRate-NRate curve corresponds to the smaller value of NRate, thus indicates the better performance. As we can see in Figure 4, for both the datasets the SLP method has the lowest PRate-NRate curve.



(a) Dataset *telecom* (b) Dataset *unicom*

Figure 4. Relation between *PRate* and *NRate*.

We set $pct = \{1, 5, 10, 20, 30, 40, 50\}$ and calculate PRate and NRate for each pct on both datasets, Tables IV and V show the results. We can see that our SLP methods perform best most of the time, the only exception occurs on metric $PRate$ in dataset *telecom* when $pct = 1$.

Table IV
TOP RANK USERS ANALYSIS ON DATASET *Telecom*.

	SLP		SLP(fix label bias)		SLP(without influence)		SVM_{node}		$SVM_{node+edge}$	
pct	NRate	PRate	NRate	PRate	NRate	PRate	NRate	PRate	NRate	PRate
1	**3.58E-04**	0.011000276	0.001218638	0.013619914	5.02E-04	0.013678087	0.001146953	0.012338569	0.001935484	**0.017171244**
5	**0.001146953**	0.092477573	0.002078853	**0.121237255**	0.002293907	0.116344264	0.004516129	0.049299164	0.006666667	0.063500505
10	**0.002867384**	0.208878173	0.004874552	**0.249578396**	0.005734767	0.245385934	0.012473118	0.092255289	0.016200717	0.120657359
20	**0.00953405**	**0.428212547**	0.012759857	0.419948563	0.018853047	0.402077095	0.031182796	0.185969505	0.043225806	0.230139004
30	**0.021218638**	**0.586701571**	0.029892473	0.569474909	0.041863799	0.531116622	0.065304659	0.297436086	0.077562724	0.348879704
40	**0.051326165**	**0.70660635**	0.05734767	0.685530755	0.07562724	0.643393344	0.109605735	0.431073145	0.123727599	0.478905729
50	**0.108100358**	**0.802287744**	0.110250896	0.783370993	0.131971326	0.747166958	0.177060932	0.571574967	0.190681004	0.613291693

Table V
TOP RANKING USERS ANALYSIS ON DATASET *Unicom*.

	SLP		SLP(fix label bias)		SLP(without influence)		SVM_{node}		$SVM_{node+edge}$	
pct	NRate	PRate	NRate	PRate	NRate	PRate	NRate	PRate	NRate	PRate
1	**1.58E-04**	**0.076212833**	8.72E-04	0.072613459	6.34E-04	0.072300469	0.001981611	0.03943662	0.00245667	0.038497653
5	0.004200186	0.310328638	0.004279425	0.313771518	**0.004120821**	**0.31799687**	0.015454431	0.161189358	0.01482102	0.130672926
10	**0.011808291**	**0.449608764**	0.01323485	0.441940532	0.013789274	0.44084507	0.03851571	0.262754304	0.032969713	0.224100156
20	**0.041845018**	**0.650391236**	0.043429427	0.628169014	0.048422761	0.621126761	0.101916836	0.417370892	0.0863887	0.381533646
30	**0.09454733**	**0.799530516**	0.095497573	0.79029734	0.117927704	0.767918623	0.201458632	0.549608764	0.164059568	0.517683881
40	**0.18671899**	**0.897026604**	0.1897287	0.892175274	0.217865067	0.867762128	0.355765854	0.658841941	0.268602495	0.633176839
50	0.316849546	0.95086072	**0.311934571**	**0.951486698**	0.328342138	0.934741784	0.537014565	0.751330203	0.391055392	0.731455399

VII. RELATED WORK & CONCLUSION

Previous works on profiling social users mainly focus on discovering user interests to provide personalized search result [10], [11], news recommendation [12], targeted advertisement [13], [14], inferring user home location [15], and inferring college, major and so on [4]. Our work infers users' affiliation in the context of social media.

This study is mainly inspired by the recent works on graph-based learning[6], [16]. [6] proposes supervised random walk to learn the edge weights for link prediction in social network. [16] extends the model for tag recommendation with multi-type edges and nodes. Our study is different from [6], [16]: 1) our problem is to predict the class label of any node in the network while their problems are to predict the similarity distance between two nodes; 2) the variable of influence bias, which improve the prediction performance, is newly introduced in our model. The modeling of node influence here is different from the previous works on the general social influence, it models the influence to attract the working colleagues as her social friends.

In this paper, we formulate the problem of inferring users' affiliation in the context of social media, and transform it as a task of classifying nodes over graphs. Then, we propose the supervised label propagation model to address this problem. This model provides a uniform way to combine all the features on the social activities. Experimental results show that our model outperforms the compared baseline methods. With identified employees, we will monitor their tacks on certain business area in social media in the future work, and show how these tracks help in business intelligence.

ACKOWNLEDGEMENTS

The authors Guangxiang Zeng and Enhong Chen were supported by grants from Natural Science Foundation of China (Grant No. 61073110), Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20113402110024), and National Key Technology Research and Development Program of the Ministry of Science and Technology of China (Grant No. 2012BAH17B03).

REFERENCES

[1] E. Keller and J. Berry, *The influentials: One American in ten tells the other nine how to vote, where to eat, and what to buy*. Free Press, 2003.

[2] G. Weimann, *The influentials: People who influence people*. SUNY Press, 1994.

[3] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: quantifying influence on twitter," in *WSDM '11*, 2011, pp. 65–74.

[4] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel, "You are who you know: inferring user profiles in online social networks," in *WSDM '10*, 2010.

[5] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," in *ICML '06*, 2006.

[6] L. Backstrom and J. Leskovec, "Supervised random walks: predicting and recommending links in social networks," in *WSDM '11*, 2011, pp. 635–644.

[7] A. Andrew, "Convergence of an iterative method for derivatives of eigensystems," *Journal of Computational Physics*, vol. 26, pp. 107–112, 1978.

[8] C. D. Brown and H. T. Davis, "Receiver operating characteristics curves and related decision measures: A tutorial," *Chemometrics and Intelligent Laboratory Systems*, vol. 80, pp. 24–38, 2006.

[9] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[10] F. Qiu and J. Cho, "Automatic identification of user interest for personalized search," in *WWW '06*, 2006, pp. 727–736.

[11] S. Xu, S. Bao, B. Fei, Z. Su, and Y. Yu, "Exploring folksonomy for personalized search," in *SIGIR '08*, 2008, pp. 155–162.

[12] R. W. White, P. Bailey, and L. Chen, "Predicting user interests from contextual information," in *SIGIR '09*, 2009, pp. 363–370.

[13] F. Provost, B. Dalessandro, R. Hook, X. Zhang, and A. Murray, "Audience selection for on-line brand advertising: privacy-friendly social network targeting," in *KDD '09*, 2009, pp. 707–716.

[14] A. Ahmed, Y. Low, M. Aly, V. Josifovski, and A. J. Smola, "Scalable distributed inference of dynamic user interests for behavioral targeting," in *KDD '11*, 2011, pp. 114–122.

[15] R. Li, S. Wang, H. Deng, R. Wang, and K. C.-C. Chang, "Towards social user profiling: unified and discriminative influence model for inferring home locations," in *KDD '12*, 2012, pp. 1023–1031.

[16] W. Feng and J. Wang, "Incorporating heterogeneous information for personalized tag recommendation in social tagging systems," in *KDD '12*, 2012, pp. 1276–1284.