Extracting User's Hidden Profile on Twitter

Dong Wang, Mohan Yang, Yuchen Liu
Department of Computer Science
University of California, Los Angeles
{dongw, yang, yliu}@cs.ucla.edu

ABSTRACT

In a social network environment like Twitter and Facebook, user modeling is an essential approach to know a user's interests, and thus making other additional services such as recommendation system possible and accurate. In user modeling, user profile is an important factor since it's written by user and it provides pretty accurate information. Previous social network websites such as Facebook and MySpace build a detailed profile when a user starts to use, letting a user fill in his/her personal information, education and work, family information as well as interests. However, Twitter has a different setting for user profile and it only allows a user to fill in a short bio within 160 characters to describe him/herself. Without the structured information, it's getting harder to build an accurate user profile. Meanwhile, according to our statistics, about 27.2% of users do not have a self-written bio or their bio is less than 5 characters and about 43.9% of users don't have a meaningful bio description since there are less than 10 characters in their bio. Thus, it becomes important for us to extract useful user profile for those who don't have a meaningful bio in order to model users accurately. In this paper, we propose approaches and models in two different categories, using social network link structure only and using user-generated content only, respectively. We also propose a co-training model to combine approaches in these two categories to make use of their advantages to achieve a better performance. In our UCLA dataset, co-training model performs a 87% precison at top 50 results, which give out a promising results for extracting users' hidden profile problem.

1. INTRODUCTION

Twitter, a micro-blogging system combining social network and text content, has demonstrated itself as a leading breaking news provider, and a platform of sharing opinions and interests. Currently, according to Twitter Blog[1], there are about over millions of users publishing more than 200 millions of tweets on Twitter every single day. Although there are existing successful social networks websites before Twitter comes out, like Facebook, MySpace etc., Twitter becomes popular because of its simplicity and people find it easy to tweet and share information. There are several characteristics which differentiate Twitter with other social networks:

- **Non-Reciprocal Relationship:** In Twitter, users are connected with links called ”follow”. If user A is following user B, all tweets which are produced by B will appear in user A's timeline. However, this ”follow” link does not require user B's permission and also B is not required to follow back. This characteristic means a user could follow anyone he/she is interested in freely. In Facebook or MySpace, users are connected by ”friendship” and this link relationship is reciprocal meaning that two users need to agree with each other to be in such a ”friendship”.
- **Incomplete Profile:** As Facebook provides a detailed profile covering personal information, family, education and work, interests, Twitter provides a simple bio information for user to describe him/herself. However, the bio information in Twitter is limited by 160 characters and does not contain any structured format. This short bio will make it difficult to know a user's profile, including affilication, occupation, interests etc., and thus hard for us to make more services, such as recommendation.
- **User-generated Content as Plain Text:** While Facebook allows users to post different types of user-generated contents, such as status, photos, videos and external urls, Twitter only has one form of user-generated content - plain text data within 140 characters (so-called tweet). However, users could insert shortened urls of an image, a video or a webpage into tweets. This allows us to analyze users' tweets easily with only text data but also make it hard to target users' interests in a specific area, like music or movies.

User profiles act as an important factor in user modeling since the profile is completed by users and often is of good quality. With an accurate user modeling, many useful applications could be accomplished such as recommendation systems, advertisement delivery, information prioritization. However, as mentioned above, the design of short bio in Twitter makes it difficult to know a user's real information. According to our experiment, about 27.2% of users don't have a self-written bio or their bio is less than 5 characters.

About 43.9% of users have a bio less than 10 characters, which indicates that almost half of users don't have a meaningful bio to describe themselves. In order to cope this problem, we could make use of the structure of the social network as well as the content information in users' tweets to extract such hidden profile for those who don't have a informative bio because they are lazy or they don't want to post those information. For example, if a user is studying at UCLA, even though he/she might not explicitly write this fact in his/her bio, we may find the following observations from his/her activities in Twitter:

1. In his/her social networks, including users he/she is following and users who is following him/her. there might be a considerate amount of users who is explicitly indicating they are students at UCLA.
2. In his/her tweets, she/he might post about what's happening in UCLA or something related to UCLA which could infer that she/he is a student at UCLA. She/he could also retweet tweets containing such information.

Based on above observations, in this paper, we explore models and approaches to extract user's hidden profile from what they are connected in the social networks and what they are tweeting on Twitter. Specifically, on a social network graph, given a category and a set of seed users in the category, we want to find users in the graph who is likely to belong to this category and give out a ranking of users based on how likely they are to be in this category. In our solution, the approaches could be divided into two categories: 1) exploiting link structure in the graph only; 2) exploiting user-generated content information only, including bio, location and tweets. After describing three models in the above two categories, we also propose a co-training algorithm to combine the advantages of models in both categories to achieve a better performance with iterative reinforcing the result of one algorithm by the result of the other algorithms. With the help of information from both perspectives, we are achieving a 87% result at top-50 precison metric for UCLA dataset.

The rest of the paper is organized as follows. Section 2 presents the mathematical formulation of the problem. Section 3 presents four solutions to this problem. Section 4 reports the experiment results. Section 5 discusses some problems discovered in the experiment. The paper concludes in Section 7.

2. PROBLEM FORMULATION

We formulate the social network in Twitter as a directed graph $G = (V, E)$. Each node $u \in V = \{1, 2, \cdots, n\}$ represents a user, and a directed edge $(u, v) \in E$ indicates user u is following user v. The set of users who are following user u is denoted as $follower(u)$, and the set of users followed by u is denoted as $following(u)$. The size of these two sets are $|follower(u)|$ and $|following(u)|$, respectively.

Given a category $\mathcal{C}$, we want to identify all the users that belong to $\mathcal{C}$. For example, if the category is "people in UCLA", the task is to identify all the users in UCLA. The user set V is partitioned into two disjoint sets $\mathcal{A}$ and $\mathcal{B}$ based on some prior knowledge. It is almost for sure that the users in $\mathcal{A}$ belong to $\mathcal{C}$, whereas the results for users in $\mathcal{B}$ are unknown. For example, users in V can be partitioned based on whether keyword "UCLA" appears in a user's biography or not. It is clear that users in $\mathcal{A}$ are highly likely to be UCLA-related. However, a user in $\mathcal{B}$ may still belong to $\mathcal{C}$ even if he does not have "UCLA" is his biography. A student who writes "loving being a BRUIN and loving God!" in his biography belongs to set $\mathcal{B}$, but he is actually a UCLA student.

Our problem is to predict how a user $u \in \mathcal{B}$ is related to the category $\mathcal{C}$ given the user partition $\mathcal{A}$ and $\mathcal{B}$. We focus on the scenario where the prior knowledge is a keyword $z_{\mathcal{C}}$ which describes the category $\mathcal{C}$. Each user u is related to a relevance score s_u which represents the likelihood that u belongs to $\mathcal{C}$. The higher the value of s_u, the more likely u belongs to $\mathcal{C}$. The result is a ranking of users in $\mathcal{B}$ based the value of s_u.

3. OUR APPROACH

3.1 Snowball Algorithm

Twitter user follows other users based on his own interest, while user's tweets are pushed to his followers. Given a category $\mathcal{C}$, the probability of a user u belonging to $\mathcal{C}$ is determined by the relevance score s_u. Assume the probability of u publishing a tweet belonging to $\mathcal{C}$ is also s_u, and each user publishes the same number (k) of tweets. Then the total number of tweets received by u is $|following(u)|k$, while $\sum_{v \in following(u)} s_v k$ of them belong to $\mathcal{C}$. The probability of receiving a tweet in $\mathcal{C}$ by u is given by

$$\frac{\sum_{v \in following(u)} s_v k}{|following(u)|k} = \sum_{v \in following(u)} \frac{s_v}{|following(u)|}.$$

Further assume that a user publishes exactly what he receives, then the probability of publishing a tweet in $\mathcal{C}$ by u is

$$s_u = \sum_{v \in following(u)} \frac{s_v}{|following(u)|}. \tag{1}$$

The snowball algorithm iteratively calculates the value of s_u according to Eqn. (1). Initially, s_u is set to 1 for $u \in \mathcal{A}$, and s_u is 0 for $u \in \mathcal{B}$. During each iteration, the algorithm scans through each user $u \in \mathcal{B}$, and updates s_u according to Eqn. (1) (the s_v value in equation is the old value in the last iteration). The algorithm iterates until the terminate condition is satisfied. Typical terminate condition can be an upper bound on the iteration number, or the ranking of users does not change during two consecutive iterations. The users in $\mathcal{B}$ are ranked based on the final value of s_u.

3.2 Bidirectional Snowball Algorithm

The snowball algorithm considers the tweets propagation from user to his followers. The inverse part of this process is the propagation from a user to his followings. A user in $\mathcal{C}$ tends to follow many users in $\mathcal{C}$, while a user followed by many users in $\mathcal{C}$ tends to belong to $\mathcal{C}$. This intuition leads to the bidirectional snowball algorithm. It calculates two vectors, $P = (p_1, \cdots, p_n)$ and $Q = (q_1, \cdots, q_n)$, where p_u represents the relevance score of user u to category $\mathcal{C}$ with respect to the following direction, and q_u represents the relevance score with respect to the follower direction.

The updating process is similar to the snowball algorithm:

$$p_u = \sum_{v \in following(u)} \frac{p_v}{|following(u)|}$$
$$q_u = \sum_{v \in follower(u)} \frac{q_v}{|follower(u)|} \qquad (2)$$

The initial value of p_u and q_u is 1 for $u \in \mathcal{A}$, otherwise it is 0. Each iteration updates the value of P and Q based on Eqn. (2). Finally, the users are ranked according to the relevance score $s_u = p_u q_u$, which is a combination of relevance to category $\mathcal{C}$ from both following and follower directions.

3.3 Naive Bayes Algorithm

The above two algorithms works well in many situations without using any tweets information. Our third approach builds a classifier using the tweets information, and predicts how likely a user in $\mathcal{B}$ belongs to category $\mathcal{C}$. The users in $\mathcal{A}$ are positive training examples, and the users in $\mathcal{B}$ are negative training examples.

There are many machine learning algorithms which address this classification problem. We use the naive Bayes classifier [8] which is one of the most effective and efficient classification algorithm. The classifier separates the users into two classes, users belong to $\mathcal{C}$ $(c = 1)$ and users do not belong to $\mathcal{C}$ $(c = 0)$.

For each user u, let T_u be the collection of u's tweets, location and biography. The word set for corpus $\bigcup_{u=1}^{n} T_u$ is $W = \{w_1, \cdots, w_m\}$. $\mathbf{1}_{T_u}(w_i)$ is the indicator function of T_u, where $\mathbf{1}_{T_u}(w_i)$ equals to 1 if word w_i appears in T_u, otherwise it equals to 0. Let $w_i = 1$ represents the event that word w_i occurs, and $w_i = 0$ represents the event that word w_i does not occur.

For each user u, the naive Bayes classifier tries to find $i \in \{0, 1\}$ which maximizes

$$p(c = i | w_1 = \mathbf{1}_{T_u}(w_1), \cdots, w_m = \mathbf{1}_{T_u}(w_m)),$$

or equivalently maximizes

$$p(c = i) \prod_{j=1}^{m} p(w_j = \mathbf{1}_{T_u}(w_j) | c = i).$$

As the problem has only two classes, it is equivalent to determining the sign for s_u:

$$s_u = \log(p(c = 1)) - \log(p(c = 0))$$
$$+ \sum_{j=1}^{m} (p(w_j = \mathbf{1}_{T_u}(w_j) | c = 1) - p(w_j = \mathbf{1}_{T_u}(w_j) | c = 0)).$$

Standard naive Bayes classifier determines whether u belongs to $\mathcal{C}$ based on s_u is positive or negative. Instead of using the sign of s_u, we use the value of s_u to determine how likely u belongs to $\mathcal{C}$. The larger the value (positive value) of s_u, the more likely u belongs to $\mathcal{C}$; the smaller the value (negative value), the more likely u does not belong to $\mathcal{C}$. Finally, the users in $\mathcal{B}$ are ranked based on the value of s_u.

The parameters of the classifier is determined by counting. The value of $p(c = 1)$ is calculated as the fraction of positive training examples, and the value of $p(w_j = \mathbf{1}_{T_u}(w_j) | c = 1)$ is calculated as the number of positive examples such that w_j appears (does not appear) in T_u divided by the number of positive examples if $\mathbf{1}_{T_u}(w_j) = 1$ (if $\mathbf{1}_{T_u}(w_j) = 0$). The calculation of $p(c = 0)$ and $p(w_j = \mathbf{1}_{T_u}(w_j) | c = 0)$ are similar.

3.4 Co-training Algorithm

The bidirectional snowball algorithm and naive Bayes algorithm focus on different perspective of twitter networks: the former focus on the network structure level, and the later focus on the tweets information level. Both of them provide reasonably well rankings for the training data, while the perspectives of these two algorithms are conditionally independent given the ranking. A more powerful approach called co-training [2] combines the two algorithms together, and iteratively reinforce the result of one algorithm by the result of the other algorithm.

Co-training algorithm is described in Algorithm 1. Users are partitioned into two disjoint sets $\mathcal{A}$ and $\mathcal{B}$ described in Section 2. l and k are positive integer parameters, and $k > l$. During each iteration, top l users generated by the naive Bayes algorithm and top l users generated by bidirectional snowball algorithm are merged into set $\mathcal{A}$. The merged users are considered as positive training examples during later training process. The iteration is executed until the top k users generated by the bidirectional snowball algorithm and naive Bayes algorithm are the same. The result of the last execution of the naive Bayes algorithm is used as the final result of the co-training algorithm.

Algorithm 1 CO-TRAINING

Input: Category $\mathcal{C}$, two disjoint sets $\mathcal{A}$ and $\mathcal{B}$, parameter k and l

Output: An array $rank$ containing users in $\mathcal{B}$ ranked on the probability of belonging to $\mathcal{C}$

```
repeat
    $rank' \leftarrow$ bidirectional snowball algorithm($\mathcal{A}$, $\mathcal{B}$)
    $rank \leftarrow$ naive Bayes algorithm($\mathcal{A}$, $\mathcal{B}$)
    $\mathcal{A} \leftarrow \mathcal{A} + \{$top $l$ users in $rank'\}$
    $\mathcal{A} \leftarrow \mathcal{A} + \{$top $l$ users in $rank\}$
until Top $k$ users in $rank'$ and $rank$ are the same
return $rank$
```

4. EXPERIMENTS

In this section, the ranking performance of our proposed four methods, including snowball algorithm (SA), bidirectional snowball algorithm (BSA), naive Bayes algorithm (NBA) and co-training algorithm (CA), is tested on real dataset extracted from twitter. The precision of top k results (precision@k) and recall of top k results (recall@k) are used to evaluate the ranking result. Precision@k (Recall@k) is the ratio between the number of users that belong to $\mathcal{C}$ in top k results and k (total number of users in $\mathcal{C}$ in the dataset). As the value of recall@k is proportional to the value of precision@k, the experiment only shows the value of precision@k.

4.1 Experiment Setup

Data collection: Twenty seed users were selected from each of the four universities, including UCLA, USC, Stanford and MIT. User profile (id, location, screen name, number of followings and followers, and a short biography) of these seed users was crawled using the Twitter API. Starting from these seed users, a two-level breadth first traversal was performed to retrieve their followers and their followers'

followers. If a new user from these four universities was discovered during the procedure (the user's biography contains the university name), this user was marked as a seed user and another two-level breadth first traversal starting from this user was performed. The data was collected during October, 2011. The crawler had collected more than 540,000 users and 780,000,000 following relations. It had also collected the most recent 20 tweets for each user, summing up to $15,321,508$ tweets in total. These tweets contain $3,143,115$ different words. After eliminating the words which appear less than 100 times, there are about $20,000$ words left.

Data preprocessing: Experiment tested different algorithms' ability on discovering users from UCLA, USC, Stanford and MIT. Users in each university were considered as a category. For example, the users in UCLA belongs to category UCLA, and the keyword $z_{\mathcal{C}}$ is the university name "UCLA". For each category $\mathcal{C}$, the users were partitioned into two sets $\mathcal{A}$ and $\mathcal{B}$ based on whether $z_{\mathcal{C}}$ appeared in their biographies. We randomly selected 20% users in $\mathcal{A}$, removed the biography of these users, and moved them to $\mathcal{B}$.

Data labeling: Top 100 results of BSA, NBA and CA from UCLA category are manually labeled. The labeling process considers the user's biography, location, tweets, following relations, and the search result from Google using the user's name as keyword. This process tries to discover as many UCLA users as possible.

4.2 Ranking Performance

This part of experiment evaluated the precision@k for different algorithms. The algorithm worked on the processed user sets $\mathcal{A}$ and $\mathcal{B}$, while the evaluation is done on the original dataset. A user which has keyword $z_{\mathcal{C}}$ in his biography belongs to $\mathcal{C}$, otherwise he does not belong to $\mathcal{C}$. The precision@k curves of different methods on different categories are shown in Figure 1.

From Figure 1 we can see that the NBA and CA performs the best at almost every positions. This fact due to the reason that the label data is automated generated by computer. User without category keyword or without large amount of related word will be negative examples in the evaluation. Compare NBA and CA, opposite to our expectation that NBA performs better than CA that is because BSA seriously affected the accuracy of training labels in CA in our computer evaluation system.

The BSA performs better than SA in MIT and USC category, and comparable with SA in Stanford category. But it performs worse than SA in UCLA category, especially in top positions. When we look insight the ranking result, we find that BSA is not as bad as it shows in the figure. For example, the user ranked at the third place in UCLA category is "LittleBigginKip", you will find he is a UCLA football team member as soon as you saw his twitter page. However, his profile didn't mention that he is a UCLA student at the time we crawl the data. Moreover, note that much of the improvement in the precision curve in MIT and USC category come in the area after top 10 positions. This is the area that we most care about, since that users in top position have some features that is easy to discover and users in these middle or tail positions reflect the effectiveness of our algorithm more clearly.

In addition to the keyword based evaluation, Figure 2 represents the precision curve in UCLA category on human labeled data. The NBA and CA still performs the best, which shows that users ranked at the top 10 positions by these algorithms are 100% belong to category UCLA. The BSA performs better than SA since that BSA penalty the users that only interesting to our target category but not actually belong to our target category. For example, the user "openwestwood" follows a lot of UCLA stuff which ranked at the 8th position in SA result but after top 100 positions in BSA result. Actually, after human labeling the experiment result, we find many interesting phenomenons and we will discuss them in our discussion section.

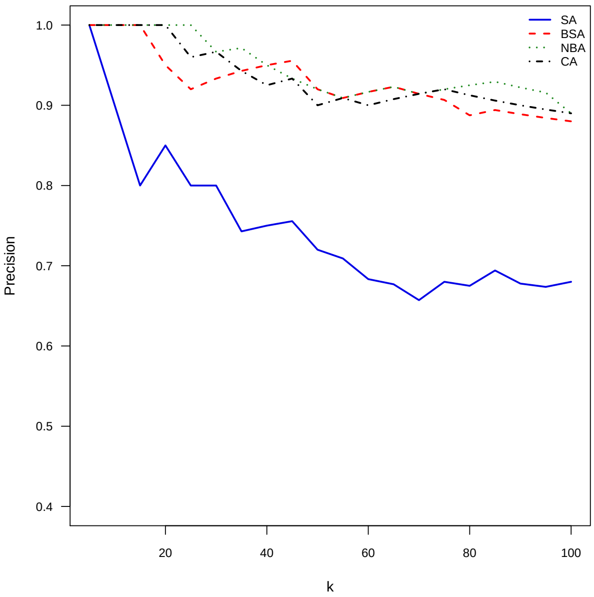


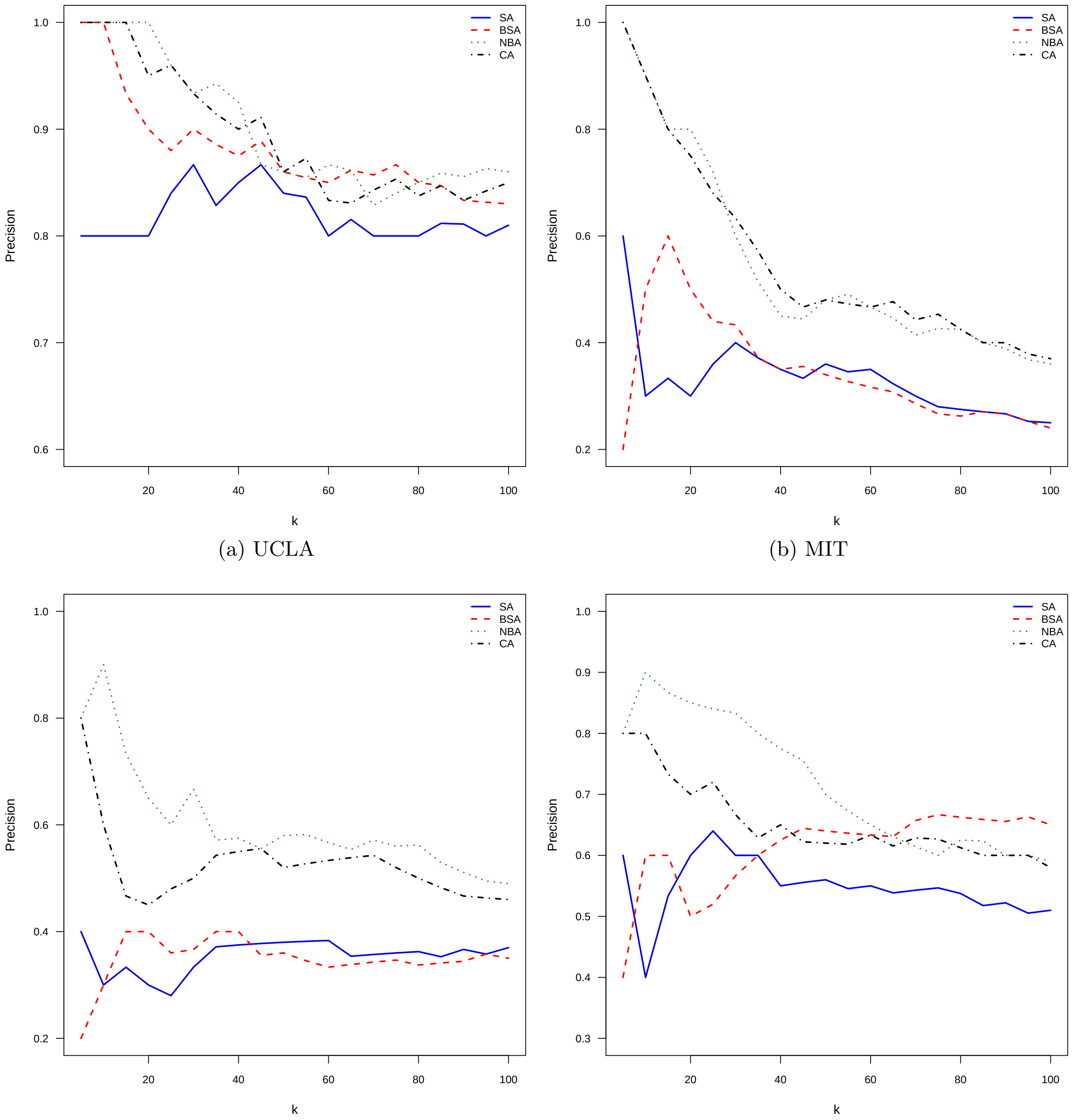
Figure 2: Precision@k for UCLA category on human labeled data

Moreover, we show some important features generated by NBA in Table 1. The words are ranked according to their occurrence $p(w_i = 1|c = 1)$, and top 10 words in each category are shown in the table. The top word features for different category are completely different. For example, UCLA students like "dailybruin" news, call themselves as "bruins" and lived near "westwood"; USC students like tweets from "usc annenberge" and the idol statue in their university is "Trojan"; MIT students live in "Cambridge" and there is famous "media lab" in their computer science department; Stanford students are glad to talking about their athletic team using nick name "cardinal".

4.3 Ranking with Loss of Information

In this part of the experiment, the stabilities of different algorithms were tested with loss of information. We randomly selected $1 - t\%$ users in $\mathcal{A}$, removed the biography of these users, and moved them to $\mathcal{B}$. The performance of different algorithms under different t is tested, and precision@k curve is shown in Figure 3. SA was not test in this experiment, since its performance is poor in the previous experiment on human labeled data.

The results show that with the loss of information, the performance of BSA falls down dramatically. It also suggests that with few positive training examples, training users in UCLA becomes less connected. The poor graph structure has a bad impact on the performance of BSA. The NBA still



(a) UCLA

(b) MIT

(c) Stanford

(d) USC

Figure 1: Precision@k for different categories

UCLA	dailybruin, bruin, bruins, westwod, neuheisel, alumna, wooden, undergraduate, midterm, royce
USC	ascj, uscedu, annenberg, uscpsycho, uscannenberg, ausc, beattheirish, trojan, trojans, atrojan
MIT	sloan, medialab, cambridge, joi, kayak, bostonupdate, alums, mechanical, techreview, edu
Stanford	stanfordfball, gostanford, astanford, gsb, cardinal, cantor, tristanwalker, auditorium, alums, freshmen

Table 1: Top 10 words from user's biography, location and tweets in different universities

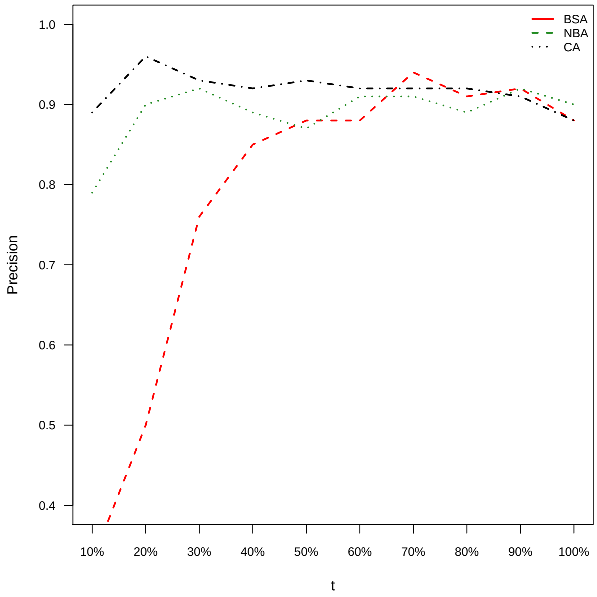


Figure 3: Precision@k for UCLA category with loss of information

have a very good result with 20% positive training examples. Within those percentage of training users, we could already detect the topic among the tweets published by users belongs to UCLA category. When we look inside the ranking result, the top users returned by BSA and NBA are very accurate. So that the amount of positive training examples does not affect the ranking result of CA very much.

Noticed that sometimes with less positive training examples, our algorithms could achieve better prediction results. That is because when we erase keyword in user's profile, the user is still in our dataset. The total number of users belong to UCLA category in evaluation data increased. So that the precision for top 100 users will be higher.

In our original dataset, the training users are likely to connected with each other since they are in same university. However, for other categories such as father, gamer or phd, users belong to such categories maybe not likely to connected with others in the same category. When we erase the profile keyword in label data, the connection in label data become smaller and the graph structure may like these categories. The performance of our algorithms on UCLA category also demonstrate that our algorithms could be applied to different categories.

5. DISCUSSION

The manually data labeling process revealed several interesting problems. The top 100 users of UCLA category returned by different algorithms were classified into several categories and described phenomenons as follows.

5.1 User Classification

The users are manually classified into 5 categories, including active users in UCLA with related biography, tweets or photos (ActiveUCLA), inactive users in UCLA (InactiveUCLA), inactive users not belonging to UCLA (Inactive), user intent miss match (Intent), and active user not belonging to UCLA (Negative). The classification result is shown in Figure 4.

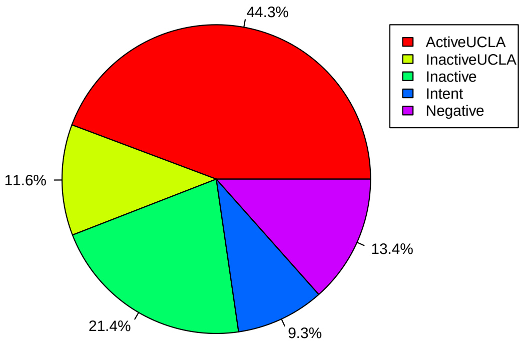


Figure 4: Manually user classification

For active users, it is easy to see whether the user has related tweets about UCLA or photos about UCLA. For users without new tweets in recent two months, we call them inactive users and predict whether they are belong to UCLA category according to search result in Google. We type their names with UCLA and see whether there is related search results in top 10 positions.

5.2 Intent Mismatch

The intent mismatch, labeled as "Intent" in Figure 4, refers to the situations where the users are somehow interested in UCLA but it is actually not belong to UCLA. The data show that intent mismatch often arises when a user follows a lot of UCLA users or a user talks about something that mentions UCLA.

There are several examples of intent mismatch. First, users may follow UCLA members in order to have more business opportunities, such as "WeTutorLA" and "bombaybite". The former one follows lot of students from UCLA, USC, UCSD and so on to let them noticed. The latter one follows a lot of organizations of UCLA because it is a restaurant near UCLA. Second, some users keeps follow back a lot of users, such as "0neNiteStan". This kind of users have a lot of tweets for advertisement. Third, users like "USCTrojansNews", "openwestwood" would ranked at higher position in our result. These users' tweets have significant intersection with tweets generated by users from UCLA. For example, "USCTrojansNews" often publish tweets that compare UCLA with USC while some users in UCLA category also like to do that. This makes our model make mistakes during the training process.

These kinds of intent mismatch contribute to 9.3% discrepancies in the data. Typically, intent mismatch are very hard to be corrected since it requires human understanding of why this user may related to target category but not belongs to that category.

5.3 Precision of Active User

Noticed that there are about 30% users are inactive users. Those users published several tweets after they register and didn't publish any more tweets during recent three months. In real cases, we don't want to see these inactive users since we don't expect that they will provide more information or business opportunity in the future. So that we evaluate our different approaches with inactive user as a negative example and the result is in Figure 5.

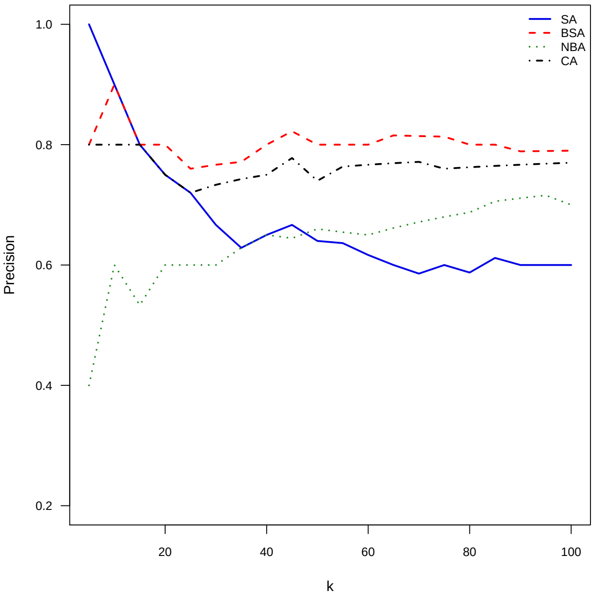


Figure 5: Precision@k of active users in UCLA category

Unexpected that SA has the best result and NBA performs the worst in top positions. We look at top users returned by NBA, and found that users such as "bruinmarketing" ranked at higher positions by NBA. It seems that "bruinmarketing" is in UCLA category, but this user has no tweets among this year. Similarly, some users only mentioned word "bruin" in their short bio and ranked higher in NBA turns out to be inactive users.

Moreover, we think that there still exists some disadvantages to treat our task as a classification problem. One disadvantage is that the class is unbalanced. The fraction of positive training samples is too small, which makes the learning particularly difficult. The other disadvantage is the number of features is very large, which is due to the diversity of words user used in their tweets, frequent appearance of typos and hyphen marks.

6. RELATED WORK

Most research works on Twitter investigate the network structure and the spread of information. Weng J. et al. purposed Twitter rank [5] examined the topic-specific influential on social network service. Welch, M.J. et al. conduct analysis on retweet links shows that the transitivity of topic relevance is better preserved over retweet links than general following relationship [10]. Wu, S. et al. studied the homophily between users within same categories, such as celebrities listen to celebrities, while media listen to media [11]. Other researches focus on analysing large amount of tweets. Kwak et al. conduct experiments that analysis topic trending in Twitter [7]. They discovered that most tweets posted everyday are related to news and hashtags are good indicators to detect events and trends.

The research work [11] shows that users within same categories are likely to follow each other. Similar, Java A. et al. in [6] present their observations of the microblogging phenomena find that users talk about their daily activities, share information, and connect others with similar intentions. Beside Twitter, Yang S.H. shows that information contained in interest networks and friendship networks is highly correlated in other social network services [12]. Given a user's tweets information and friendship, build a system that reconstruct his profile or interesting is very helpful for personalize information access and advertisement targeting.

One way for user profile reconstruction is calculating how user related to a given category in the graph. It is similar to computing node proximities in large graphs. There are some related works use random walk as their basic model. Tong H. et al. presented algorithm that find nodes in a centerpiece subgraph in [4]. The author also presented algorithm that compute how closely related are two nodes in a graph in [9]. Other solutions aim learning to rank nodes for target category. The most recent work [3] conducted by Backstrom L. introduced supervised random walks that combines graph structure knowledge and link level attributes to rank the users for friends recommendation.

7. CONCLUSION

Effectively estimating user profile and accordingly recommending service or suggesting friends are fundamental to all social networks. In this report, we have shown that the user's short bio is highly related to user's friendship and user's tweets. We presented three simple ranking approaches and a co-training framework that leverage both friendship and tweets evidence to solve the task purposed in our report. The graph approach analysis user profile from his followings and followers since that similar users are more likely to connected with each other. The Bayes approach extract the semantics of individual message that allow for the generation of user profile information of a given concept. Given the co-training framework, it is easy for us to combine two different approaches and obtain a better ranking result with limited positive training examples. The experiments results on twitter social network demonstrate that simple algorithms perform very well for our task. Additionally, we learned the pros and cons of different approaches from the discussion.

The co-training framework that combines the knowledge from graph structure and tweets information is not limited to predict user profiles. It can be applied to many other problems that require learning to rank nodes in a graph. There are some interest future research directions: First, the users are equally important in our model based on graph structure. However, we find many inactive users when we label the users in the ranked list. In the future, we may assume users have different weight during the training process. Thus, there may exists the underlying mechanism of how the interactions and information between users related to their personal profile. Second, it is interesting to apply our algorithms in friends recommendation system. Currently, most friends recommendation systems were based on number of users' mutual friends. The co-training approach could leverage users' mutual friends information and other user behaviors such as tweets, profiles. I think it is very helpful to build such framework for friends recommendation.

8. REFERENCES

[1] Twitter blog: 200 million tweets per day. *Available from http://blog.twitter.com/2011/06/200-million-tweets-per-day.html*, 2011.

[2] Tom Mitchell Avrim Blum. Combining labeled and unlabeled data with co-training. Conference on Computational Learning Theory, 1998.

[3] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 635–644. ACM, 2011.

[4] C. Faloutsos H. Tong and Y. Koren. Center-piece subgraphs: Problem deïňĄnition and fast solutions. In *KDD'06*, 06.

[5] J. Jiang J. Weng, E.-p. Lim and Q. He. Twitterrank: Finding topic-sensitive influential twitterers. WSDM'10, 2010.

[6] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65. ACM, 2007.

[7] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

[8] ME Maron. Automatic indexing: an experimental inquiry. *Journal of the ACM (JACM)*, 8(3):404–417, 1961.

[9] C. Faloutsos T. Tong and J.-Y. Pan. Fast randomwalk with restart and its applications. In *ICDM'06*, 2006.

[10] M.J. Welch, U. Schonfeld, D. He, and J. Cho. Topical semantics of twitter links. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 327–336. ACM, 2011.

[11] S. Wu, J.M. Hofman, W.A. Mason, and D.J. Watts. Who says what to whom on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 705–714. ACM, 2011.

[12] S.H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha. Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th international conference on World wide web*, pages 537–546. ACM, 2011.