A Novel Hybrid Friends Recommendation Framework for Twitter

Yan Zhao[1], Jia Zhu[2], Mengdi Jia[1], Wenyan Yang[1], Kai Zheng[1]

[1]School of Computer Science and Technology, Soochow University, China
[2]School of Computer Science, South China Normal University, China
jzhu@m.scnu.edu.cn

Abstract. As one of the key features of social networks, friends recommendation is a kind of link prediction task with ranking that was extensively investigated recently in the area of social networks analysis as users would like to follow people who have similar interests to them. We use Twitter as a case study and propose a novel hybrid friends recommendation framework that is not only based on friends relationship but also users' location information, which are recorded by Twitter when they posted their tweets. Our framework can recommend friends to users who have similar interests based on location features by using collaborative filtering to effectively filter out those common places which are meaningless, e.g., bus station; and focuses on those places that have high probability that people are there more likely to become friends, e.g., dance studio. In addition, we propose a multiple classifiers combination method to leverage the information contained in friends and locations features in order to get better outcomes. We evaluate our framework on two real corpora from Twitter, and the favorable results indicate that our proposed approach is feasible.

Keywords: Social Network, Recommendation Systems

1 Introduction

With the fast growing of Web 2.0, social networking sites like Twitter are becoming increasingly popular. For example, people can use Twitter to find their friends and catch up their recent status. Friends recommendation is a kind of link prediction task with ranking that was extensively investigated recently in the area of social networks analysis [5, 7, 34]. It is also very important in information management. As a typical example like Twitter, which allows its users to send and read text-based posts of up to 140 characters, known as tweets. It is not sufficient for a system to provide recommendation based on friends relationship or the most popular users because users normally only want to follow people who have similar interests [10].

[2] Jia Zhu is the corresponding author.

In addition, Twitter have millions of users that means they are often quite sparse with low density of links among users. As a result, the link prediction space is huge and highly imbalanced. Existing approaches merely focus on finding friends in the the 2-hop social neighborhood, i.e., friends-of-friends of a user [25]. It may likely result in an exponentially larger set of increasingly less likely candidates if we extend the range to 3 or more hops neighborhood. As a consequence, the friends recommendation problem appears heavily influenced by network distance between users.

To overcome this problem, we propose a friends recommendation framework to use location information that is the places visited by each user apart from friends relationship information. The increasing availability of location-acquisition technologies, e.g., GPS, nowadays enable people to log the location histories with spatial-temporal data. Such real-world location histories provide us with the correlations of users' interests and the places they have visited [1, 7, 20, 28, 33, 34, 36]. We use Twitter as a case study in this paper not only because Twitter is one of biggest social networks in the world but also the way of communication in Twitter is quite standard compared to other social networks.

However, Twitter users have the right to determine if they want to share their current location and the location information is not always available. Fortunately, we can retrieve some location information from Twitter as it added an explicit GPS tag that can be specified for each tweet in early 2010 and is continually improving the location-awareness of its service [25]. In addition, we can also extract location information from users' tweets even if users turn off the GPS tagging. For example, users who posted tweets with the keywords "step up dance studio" in the same period of time might be friends or may become friends as they all have been to "step up dance studio" or at least interested in this place. Therefore, we may recommend friends to these users based on the extracted information.

Our framework can recommend friends to users who have similar interests to them based on location features by using collaborative filtering [4, 27, 35] to effectively filter out those common places which are meaningless, e.g., bus station; and focuses on those places where have high possibility that people are there more likely to become friends, e.g., dance studio. We then pick top 10 places based on the score generated by our methods. Additionally, to achieve better recommendation outcomes, we propose a multiple classifiers combination method to combine the outputs from different classifiers. Our main contributions are summarized below:

1. We propose a hybrid friends recommendation framework, which uses collaborative filtering to effectively filter out those common places which are meaningless so that the recommendation performance can be improved.
2. We further propose a multiple classifiers combination(MCC) method that combines outputs of multiple classifiers, which leverages the information contained in the features of friends relationship and location information .

3. Extensive experiments have been performed on two real data sets we retrieved from Twitter. We show that our framework performs significantly better than baseline method in different scenarios.

The rest of this paper is organized as follows: In Section 2, we discuss related works in link prediction, particularly friends recommendation. In Section 3, we formulate our problem and describe the details of our framework. In Section 4, we present our experiments, evaluation metrics, and results. We also conclude this study and discuss future work in Section 5.

2 Related Work

Hoff et al. [11] introduced a class of latent class models from the perspective of social networks analysis which tries to project all the networked objects to a latent space, and the decision for link existence is based on their spatial positions. However, the authors have not discussed in detail about the choice of a prior distribution for latent positions so that the outcome is not really impressive. Taskar et al. [29] proposed a model by applying the Relational Markov Network (RMN) framework to define a joint probabilistic model over the entire link graph. The application of the RMN algorithm provided significant improvements in accuracy over flat classification. Though we have applied RMN and obtain success in our earlier work for link prediction [38], but we found that RMN is good at predict linking over relational data, e.g. friends relationship, but it is not suitable for uncertain relational data like location information as some relationships among objects are not useful.

Basilico and Hofmann [2] proposed to use the inner product of two nodes and their attributes as similarity measure for collaborative filtering. Their method showed how generalization occurs over pairs with a kernel function generated from either attribute information or user behaviors. They did not consider the relationship might be changed, which means the similarity measure might be different along with the time. Therefore, we do not use similarity measure for the collaborative filtering in our approach.

Huang et al. [12] introduced the time-series link prediction model problem and taking into consideration temporal evolutions of link occurrences to predict link occurrence probabilities at a particular time. We learn time-series link prediction from their model and apply additional location information into our approach.

Hannon et al. [10] focused on the creation of relationships between users and attempt to harness the real-time web as the basis for profiling and recommendation. They evaluated a range of different profiling and recommendation strategies based on a large dataset of Twitter users and their tweets. Their profiling algorithm is interesting but again they ignored those important location information which make their system recommend a list of popular persons rather than a list of persons who might have similar interest to users.

Scellato et al. [25] designed a Link prediction systems in a location-based social network called Gowalla with periodic snapshots to capture its temporal evolution. They defined new prediction features based on the properties of the places visited by users which are able to discriminate potential future links among them and found about 30% of new links are added among place-friends, i.e., among users who visit the same places. They proved the location features certainly can help people find friends though some of properties they described in the paper are not available in Twitter, which is one of the issues we try to solve in this paper.

Sadileka [23] presented a recommendation system called Flap which infers social ties by considering patterns in friendship information, the content of messages, and user location. Each component is a weak predictor of friendship alone, but combining them results in a strong model, accurately identifying the majority of friendships. They evaluated Flap on a large sample of highly active users from two distinct geographical areas and show high accuracy on the reconstruction graph even when no edges are given. However, they did not consider some places are useless, e.g. bus stop, which should not be used as location features.

Regards to extract useful location information, Zhou et al. [37] used a collaborative filtering recommenders based only on users' check-in data for location recommendation. Though their research is not for friends recommendation but their user-based model can be adapted for our approach to get common places.

Biancalana et al. [3] proposed a recommendation system to identify users' needs. Though their methodology is not for friends recommendation, but their information filtering process for points of interests are good reference for our work. Later, Gurini et al. [9] proposed a user recommendation method based on a novel weighting functions, which uses users' sentiments to build user profiles to employ in the recommendation process. Though users' sentiments are proved to be useful for friends recommendation by the authors, it can not resolve the geographical gap between users. Trattner et al.[30] recently evaluated the social proximity of users via supervised and unsupervised learning approaches and establish that location-based social networks have a great potential for the identification of a partner relationship.

3 Proposed Framework

As we mentioned earlier, our recommendation framework focuses on combining location information and friends relationship information to recommend friends to users because people who often visit the same place are normally have similar interest and likely to become friends [16].

The proposed friends recommendation framework is shown in Figure 1. We first implement a PHP script to retrieve data from Twitter by using Twitter Stream API[1], and generate a list locations based on information from GPS tags and tweets by adopting collaborative filtering methods. We then use these locations information plus friends relationship information as features to a classifier

[1] https://dev.twitter.com/docs/streaming-apis

and generate Top-K friends for users from test dataset as recommended friends according to the assignment by the classifier. There are several key components in this framework, and we will discuss each of them in the following sections.

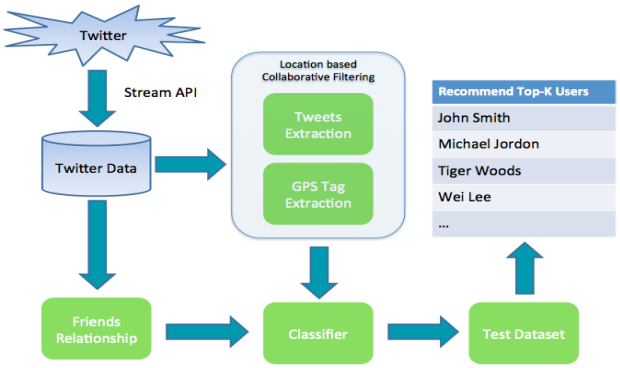


Fig. 1: Friends Recommendation Framework

3.1 Friends Relationship Features

Friends relationship also known as friends-of-friends, is always an important element to be considered in a social network friends recommendation system. Users are likely to add more friends through friends relationship as the recommended friends are users may know [17]. In the context of twitter, we define friends relationship as the common followees and followers of two users.

For example, if two users share a large amount of followers, then they are likely to have similar interests or occupation, e.g., two popular singers. It makes sense to recommend them as friends of each other to create more business opportunities. Therefore, we treat both users' followees and followers as friends of users, and we do not consider any further distance between them, e.g. 3-hop neighborhood, due to the possibility of resulting in an exponentially larger set of increasingly less likely candidates [25].

Assume a Twitter user U in the time snapshot t has a list of followees $FE(U_t)$ and followers $FL(U_t)$' union set $FR(U_t)$ as friends of U_t, we then have the recommendation score RS for user $U_t^{'}$ to be a friend of U_t as shown in Equation.(1):

$$RS(U_t, U_t^{'}) = \frac{|FR(U_t) \bigcap FR(U_t^{'})|}{|FR(U_t) \bigcup FR(U_t^{'})|} \tag{1}$$

where

$$FR(U_t) = FE(U_t) \bigcup FL(U_t) \tag{2}$$

$$FR(U_t^{'}) = FE(U_t^{'}) \bigcup FL(U_t^{'}) \tag{3}$$

We then define the friends relationship features as below:

Definition 1 *Friends Relationship Features: We call $F(U_t, U_t^{'})$ is a set of friends relationship features for a Twitter user $U_t^{'}$ at the time snapshot t to be a friend of U_t including the number of common followers and followees between users, the number of common friends(union of followers and followees), and the fraction of common friends represents as the recommendation score $RS(U_t, U_t^{'})$, which is calculated by the number of mutual friends in the dataset.*

Though friends relationship is an important factor but it is not sufficient for users to find out people who have similar interest to them because two users have many common friends may not have similar interest. In the following sections, we are going to take location information into consideration because real world location histories provide us with the correlations of users' interests and the places they have visited [36].

3.2 Location Information Extraction

There are two types of location information we are going to use from users' tweets in our framework. One is the GPS tags associated with each tweet, the other is the location information in each tweet. However, the GPS tags are not always available if users manually turn off the option or users use a laptop to post tweets. In this case, we try to explore the information in the tweets to collect possible location information. For instance, if an user posts a tweet " I am watching shows in Hilton.", then "Hilton" certainly is a meaningful location information in this context though it is possible that there are many locations with this name but we can use collaborative filtering to solve this issue. Details are given in the next section.

To extract the location related information from tweets, we used the Stanford Named Entity Recognizer[2] to perform the extraction. This recognizer provides a general implementation of Conditional Random Field (CRF) sequence models [15], which is a class of statistical modelling method to recognize the words in a text which are names, e.g. location names. CRF can take context into account compared to other ordinary classifiers which require labelling for a single sample without regard to "neighboring" samples. We choose the good 3 class (PERSON, ORGANIZATION, LOCATION) named entity recognizers for English and its performance has been described in [8].

3.3 Location Features Construction

Though we can retrieve a set of GPS tags and location names from tweets but not all of them are useful in our approach. For example, location like bus station is meaningless, people might be always in the same bus station every single

[2] http://nlp.stanford.edu/software/CRF-NER.shtml

working day but they might have never talked to each other in their whole life. Thus, the probability for them to become friends are not high from our perspective because this kind of location information (e.g., XXX bus station) does not show common interest from their tweets, particularly compared to places like "XXX dance studio". Therefore, we need to adopt collaborative filtering to effectively filter out those common places which are meaningless and focus on places that may represent users' interest. We choose two popular collaborative filtering approaches, memory-based and model-based, to perform the filtering process.

We first give a definition for the term "Common Places":

Definition 2 *Common Places: We call $SP(U_t, U_t^{'})$ is a list of common places that a Twitter user U_t shared with $U_t^{'}$ if the GPS tags in each tweet they posted at time snapshot t are in the range of 3 kilometers or location names extracted from each tweet they posted at time snapshot t are similar.*

The reason we selected 3 kilometers range as threshold is because 3 kilometers is a standard industry setting for many popular location based service mobile apps, e.g. AroundMe[3]. The location names similarity is calculated based on the popular Jaccard coefficient approach for string similarity calculation. The definition and performance of Jaccard coefficient can be found at[24].

We then have a list of features for location information:

1) *Number of Common Places* - The number of same place two users U_t and $U_t^{'}$ share at time snapshot t, represents as $|SP(U_t, U_t^{'})| = |P(U_t) \bigcap P(U_t^{'})|$ where $P(U_t)$ and $P(U_t^{'})$ are the list of places extract for U_t and $U_t^{'}$' tweets at time snapshot t [25].
2) *Fraction of Common Places* - Similar to number of common place, we have fraction of common place proposed in [25] as shown in Equation. (4):

$$|SFP(U_t, U_t^{'})| = \frac{|P(U_t) \bigcap P(U_t^{'})|}{|P(U_t) \bigcup P(U_t^{'})|}. \tag{4}$$

3) *Memory-Based Collaborative Filtering Common Places* - As we discussed earlier, some locations like bus stations are meaningless. Therefore we apply collaborative filtering to filter out these locations as collaborative filtering is a method of making automatic predictions about the interests of a user by collecting preferences from many users.
   In our case, we first used memory-based collaborative filtering [26] to calculate a score $Score_f(p)$ for each common place p of U_t and $U_t^{'}$ as shown in Equation. (5) which indicates place frequency:

$$Score_f(p) = \frac{1}{N} \Sigma_{U \in FR} K_p \cdot \frac{1}{N^{'}} \Sigma_{U^{'} \in FR^{'}} K_p^{'}, \tag{5}$$

   where N and $N^{'}$ are the number of friends, FR and $FR^{'}$ are the set of friends of U_t and $U_t^{'}$. K_p and $K_p^{'}$ are the number of times they have been to the place p at time snapshot t.

[3] http://www.aroundmeapp.com/

However, the high $Score_f(p)$ does not indicate that the location has been visited by many friends because the place may be visited by a small proportion of friends with high frequency. To address this issue, we adopt the inverse place frequency from document processing [13] as shown in Equation. (6):

$$Score_{if}(p) = log\frac{N + N^{'}}{N_p + N_p^{'}}, \tag{6}$$

where N and $N^{'}$ are the number of friends, and N_p and $N_p^{'}$ are the number of friends of U_t and $U_t^{'}$ who have visited the place p. We then have Equation. (7), which is the product of place frequency and inverse place frequency:

$$Score(p) = Score_f(p) \times Score_{if}(p). \tag{7}$$

We then pick the top 10 places with the highest score $Score_(p)$ as location features. By using this method, we can solve the issue we have addressed above as it is common sense that people will not post tweets like "I am in bus stop." many times, therefore useless places like "bus stop" will be filtered out.

4) *Model-Based Collaborative Filtering Common Places* We also use model-based clustering collaborative filtering algorithm [32] as it has been shown to be useful for classification tasks. Similar to 3), $Score(p)$ is the score for each common place p of U_t and $U_t^{'}$ at time snapshot t, but in clustering collaborative filtering algorithm, $Score(p)$ is calculated according to Equation. (8):

$$Score(p) = \sqrt[2]{\Sigma_{i=1}^{n}|x_i - y_i|^2} \tag{8}$$

where x_i, y_i are the number of friends of U_t and $U_t^{'}$ have been to the place p i times at time snapshot t, $i \leq n$. We also pick the top 10 places with the highest score as location features.

3.4 Multiple Classifiers Combination Method

Once a set of features has been obtained, we then need to choose a categorization algorithm to build a classifier that can produce the probability of two users to become friends. Note that we do not consider feature selection in this paper as it is not the main focus. Most friends recommendation algorithms are based on machine learning techniques, Support Vector Machine and Bayesian Network are two popular machine learning techniques among being widely used.

In our approach, we have four classifiers based on friends relationship and location information, namely, SVM Friends, SVM Locations, BN Friends and BN Locations. SVM and BN Friends classifiers are based on friends relationship information while SVM and BN Location classifiers use location information in each tweet as features. We did not combine friends relationship and location information features into one classifier as these two features are based on different

types of information and our preliminary analysis showed the combination of homogeneous classifiers can achieve better results which will be discussed in the coming sections.

For each classifier, We then adopt an algorithm based on the approach proposed in [25]:

For every snapshot t, we compute features for each pair of users who are not friends at t, and assign a positive label to each pair if they become friends at $t+1$, and a negative label otherwise. Thus, training and testing sets are built so that the features from a given time interval are mapped to class labels in a future time interval. Hence, given M snapshots, we can create $M-1$ learning sets, each one with labels drawn from the next snapshot. Classifiers can then be trained to build models and recognize positive and negative items from their features.

Based on this algorithm, we further propose a multiple classifiers combination (MCC) method to combine all four classifiers together because the intuition is that the combination of homogeneous classifiers using heterogeneous features can improve the final result [19].

Assume each classifier produces a unique decision regarding the recommended friends of each user U in the test dataset, we then compare the results among all the four classifiers and the final output depends on the reliability of the decision confidences delivered by the participating classifiers. We apply the concept of Decision Template (DT) to avoid the case in which the classifier make independent errors [14] and calculate the confidence score.

Assume each classifier produces a output $E_i(U) = [d_{i1}(U), ..., d_{i|G|}(U)]$ where $d_{ij}(U)$ is the membership degree given by classifier E_i for the recommended friend j to a user U in the test dataset, $j \in G$, G is the set of friends recommended or not recommended by classifier E_i. The outputs of all classifiers can be represented by a decision matrix DP, which is defined as follows:

$$DP(U) = \begin{pmatrix} d_{11}(U) & ... & d_{1|G|}(U) \\ d_{21}(U) & ... & d_{2|G|}(U) \\ d_{31}(U) & ... & d_{3|G|}(U) \\ d_{N1}(U) & ... & d_{N|G|}(U) \end{pmatrix}$$

The membership degree $d_{ij}(U)$ is calculated using the data T_f in training set, T_f indicates a person, $f = 1, 2, ..., |G|$ in each time snapshot as follows:

$$d_{ij}(U) = \frac{\sum_{j=1}^{|G|} Ind(T_j, i)}{|G|} \tag{9}$$

where $Ind(T_j, i)$ is an indicator function with value 1 if T_j is a recommended friend and 0 otherwise. At this stage, we have the membership degree for a recommended friend j to each user U and store in a matrix $DP(U)$.

We then calculate the confidence score $Score_j(U)$ for each user U using various rules from the $DP(U)$ for each recommended friend j and pick the top most recommended friends with the highest confidence score. Assume N is the number of classifiers, we apply minimum, maximum and average rules for the matrix below to consider the diversity among multiple classifiers:

$$MinimumRule : Score_j(U) = Min_{i=1}^{N}(d_{ij}(U)) \tag{10}$$

$$MaximumRule : Score_j(U) = Max_{i=1}^{N}(d_{ij}(U)) \tag{11}$$

$$AverageRule : Score_j(U) = Mid_{i=1}^{N}(d_{ij}(U)) \tag{12}$$

4 Evaluations

4.1 Corpora and Data Preparation

In our experiment, we collected public tweets from two cities by using Twitter Stream API, New York and Sydney, from March 10, 2013 to May 10, 2013. We implemented a PHP script to extract tweets that have GPS tags enabled and store into database every 15 minutes. For evaluation purpose, we only recommend friends to the users who have posted more than 5 tweets as they are active users. We also remove the users who have more than 1000 friends as they are most likely "advertisement users". We randomly selected 10% of these active users and constructed a set of pairs of users according to the algorithm we described in Section 3. The statistics of our data are shown in Table I.

Table 1: Data Corpora

	New York	Sydney
No. of Unique Users	87400	39567
No. of Tweets with Location Information	295388	157605
No. of Unique Active Users	52644	29328
Avg Tweets by Active Users	5.61	5.37
Avg Places of Active Users	7.73	7.29
No. of Followees of Active Users	3211130	2133880
No. of Followers of Active Users	4853070	2924550
No. of Pairs of Users being Evaluated	126520	49334

4.2 Evaluation Metrics and Baseline Method

We evaluated our framework based on Receiver-Operating- Characteristic (ROC) curves [22], which is the indicator adopted by most of researchers including the work on friends recommendation we introduced in Section 2.

ROC curve is a graphical plot which illustrates the performance of a binary classifier system as its discrimination threshold is varied. It is created by plotting true positive rate (TPR is also known as sensitivity) and false positive rate (FPR is also known as specificity) at various threshold settings, $TPR = \frac{TP}{TP+FN}, FPR = \frac{FP}{FP+TN}$.

In our case, TP stands for True Positive which means the number of pairs of users correctly labeled as belonging to the positive class, FP stands for False Positive which means the number of pairs of users incorrectly labeled as belonging to the positive class, TN stands for True Negative which means the number of pairs correctly labeled as belonging to the negative class and FN stands for False Negative which means the number of pairs of users who were not labeled as belonging to the positive class but should have been.

We also use the area under the ROC curve (AUC) which is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one [6]. Assume our classifiers output K positive pair of users $x_i, i = 1...KP$ and KN negative pair of users $y_j, j = 1...KN$. An unbiased estimator of the AUC is $\frac{1}{KP*KN}\Sigma_i\Sigma_j|f(\frac{(x_i)}{(i,j)}) - f(\frac{(y_j)}{i,j})|$, where function f denotes a classifier trained without the i-th and j-th training example.

For baseline method, we used the friends of friends relationship that is Twitter and other micro blog services that are currently using to recommend friends to users, e.g., Weibo[4], which is the number of common friends between users. Then the recommendation score $RS(U_t, U_t^{'})$ in baseline method is calculated as $RS(U_t, U_t^{'}) = |FR(U_t) \bigcap FR(U_t^{'})|$ derived from Equation.(1). Rather use them as features, we simply pick the top K pair of users with the highest score.

Since our approach is for friends recommendation, we randomly selected K pairs of users for each active user in individual classifiers, and picked top K pairs of users with the highest score in the MCC method. We also evaluated various cases when $K = 5, 10, 20, 30$, details are given in the following sections.

4.3 Evaluation Results

This section is to discuss the comparisons of individual classifiers as mentioned in section 3.4. All classifiers are implemented by Weka API [5], and we use RBFKernel [31] for SVM. We set $M = 5$, which means the dataset for classifiers is split to training and testing with the proportion 80% and 20%, respectively. All classifiers are trained by 5-fold cross validation.

Results of Individual Classifier We first evaluated each individual classifier by apply ROC curve on both New York and Sydney corpora with K=5, 10, 20, 30 as shown in Figures 2 and 3 from top left to bottom right respectively.

As the results shown in both New York and Sydney corpora, SVM Friends and Locations classifiers generally outperform BN Friends and Locations classifiers and the baseline method. In addition, we also notice that with the increase of number of recommended friends, the performance of all methods are dropped.

From the AUC value in Figure 4, we observe that location information alone is not sufficient to accurately provide friends recommendation though both location classifiers still perform better than baseline but not as good as friends classifiers.

[4] http://weibo.com
[5] http://www.cs.waikato.ac.nz/ml/weka/

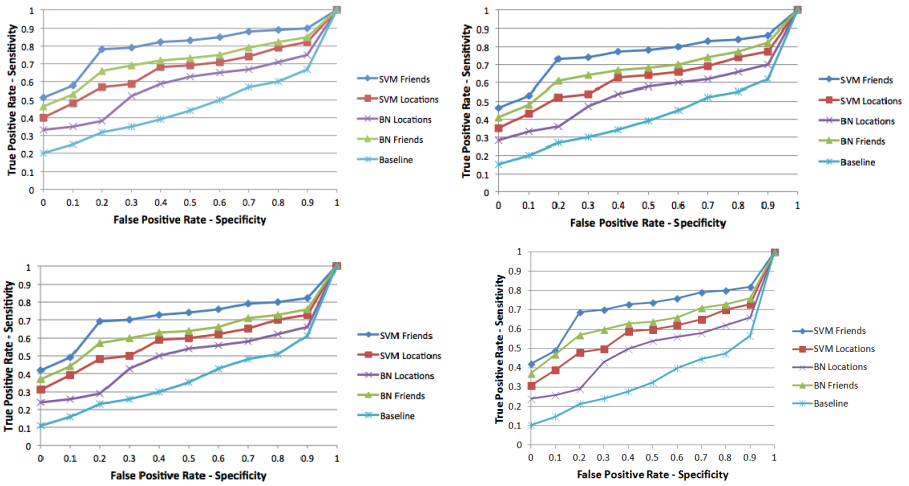


Fig. 2: ROC for individual classifier with K=5, 10, 20, 30 on New York corpus

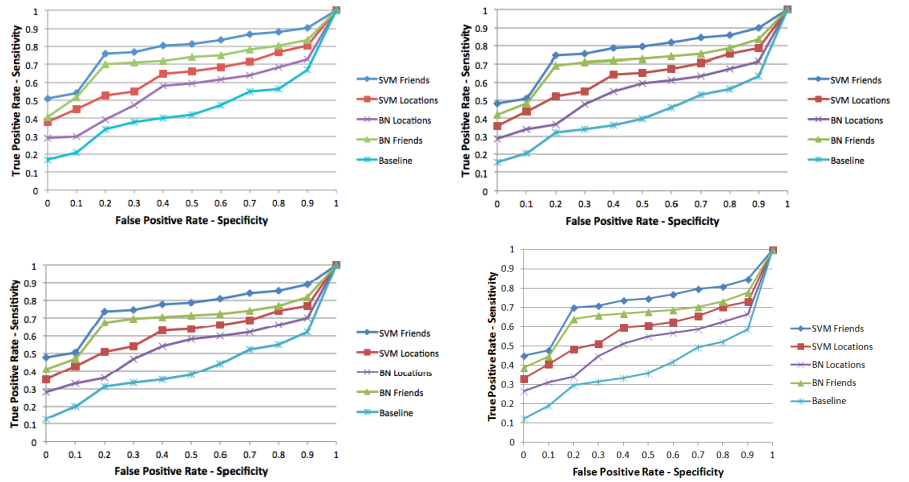
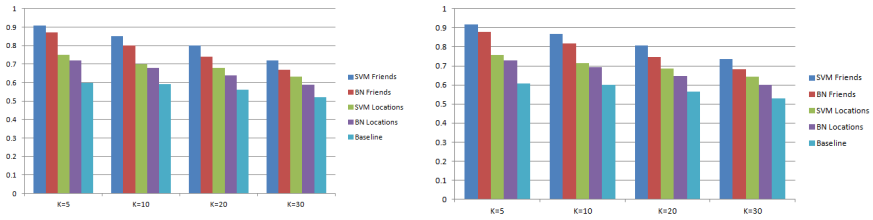


Fig. 3: ROC for individual classifier with K=5, 10, 20, 30 on Sydney corpus

Results of MCC Method In this section, we evaluated the MCC method. The main purpose of this method is to see if location information can help to improve predication accuracy compared to only use friends relationship information. Figure 5 shows the ROC curve on New York corpora with K=5, 10, 20, 30 based on various rules. Compared to the results of individual classifier, our MCC method achieve better results, particularly on average rule which proves that location features are useful for friends recommendation while keep the link prediction space as small as possible.



(a) New York (b) Sydney

Fig. 4: Overall AUC for individual classifier

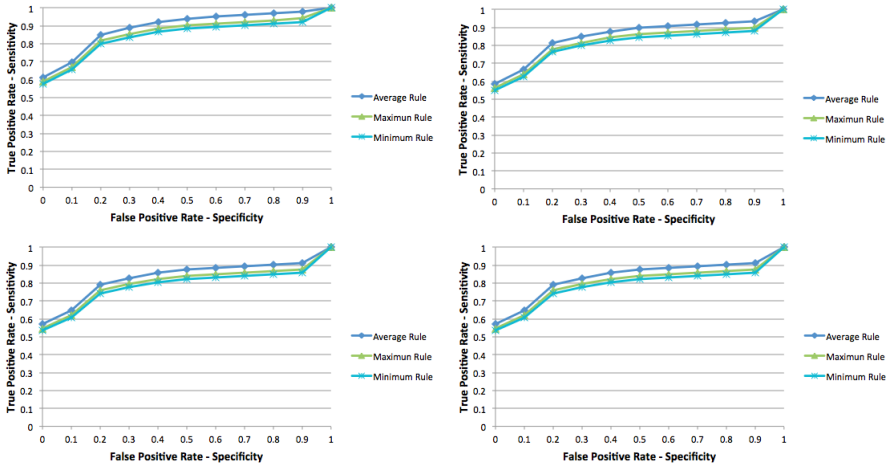


Fig. 5: ROC for MCC method with K=5, 10, 20, 30 on New York corpus

5 Conclusions

In this paper, we propose a hybrid friends recommendation framework for Twitter. Our framework not only takes existing friends relationship of Twitter users as consideration but also combines location features generated from collaborative filtering methods. In addition, we also contributed a method to extract location information from Tweets and a multiple classifiers combination method to leverage the information contained in our features, either friends relationship or locations. We evaluated our framework on two corpora from real world with comparisons between different classifiers and baseline method. The experiments indicated that our framework is feasible.

Acknowledgments

This work was supported by Natural Science Foundation of Guangdong Province, China (No.2015A030310509), and the S&T Projects of Guangdong Province

(No.2016A030303055,No.2016B030305004,2016B010109008), Natural Science Foundation of China (No. 61532018 and No. 61502324),

References

1. Bao, J., Zheng, Y., Wilkie, D., Mokbel, M.: Recommendations in location-based social networks: a survey. GeoInformatica 19(3), 525–565 (2015)
2. Basilico, J., Hofmann, T.: Unifying collaborative and content-based filtering. in ICML pp. 9–17 (2004)
3. Biancalana, C., Gasparetti, F., Micarelli, A., Sansonetti., G.: An approach to social recommendation for context-aware mobile services. ACM Trans. Intell. Syst. Technol 4(1), 1–31 (2013)
4. Bobadilla, J., Ortega, F., Hernando, A., Gutierrez, A.: Recommender systems survey. Knowledge-Based Systems 46(1), 109–132 (2013)
5. DeScioli, P., Kurzban, R., Koch, E., Liben-Nowell, D.: Best friends alliances, friend ranking, and the myspace social network. Perspect Psychol Sci 6(1), 6–8 (2011)
6. Fawcett, T.: An introduction to roc analysis. Pattern Recognition Letters 27(8), 861–874 (2006)
7. Feng, S., Huang, D., Song, K., Wang, D.: Online friends recommendation based on geographic trajectories and social relations. Advanced Data Mining and Applications 8346, 323–335 (2013)
8. Finkel, J.R., Grenager, T., C.Manning: Incorporating non-local information into information extraction systems by gibbs sampling. Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics pp. 363–370 (2005)
9. Gurini, D., Gasparetti, F., Micarelli, A., Sansonetti, G.: A sentiment-based approach to twitter user recommendation. Proceedings of the 5th ACM RecSys workshop on Recommender systems and the social web pp. 1–4 (2013)
10. Hannon, J., Bennett, M., Smyth, B.: Recommending twitter users to follow using content and collaborative filtering approaches. Proceedings of the fourth ACM conference on Recommender systems pp. 199–206 (2010)
11. Hoff, P., Raftery, A., Handcock, M.S.: Latent space appraches to social network analysis. Journal of the American Statistical Association 97(460), 1090–1098 (2002)
12. Huang, Z., Lin, D.K.J.: Time-series link prediction problem with applications in communication surveillance. INFORMS Journal on Computing 21(1), 286–303 (2009)
13. Jones, K.S.: A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation 28(1), 11–21 (1972)
14. Kuncheva, L.I., Bezdek, J.C., Duin, R.P.: Decision templates for multiple classifier fusion. Pattern Recognition 34(2), 299–314 (2001)
15. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proc. 18th International Conf. on Machine Learning. pp. 282–289 (2001)
16. Li, Q., Zheng, Y., Xie, X., Ma, W.: Mining user similarity based on location history. Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographical Information Systems pp. 247–256 (2008)
17. Liben-Nowell, D., Kleinberg, J.: The link prediction problem for social networks. In Proceedings of CIKM pp. 556–559 (2003)
18. Mitchell, T.: Machine learning. McGraw-Hill (1997)

19. Orrite, C., Rodriguez, M., Martinez, F., Fairhurst, M.: Classifier ensemble generation for the majority vote rule. Progress in Pattern Recognition, Image Analysis and Applications pp. 340–347 (2008)
20. Ozsoy, M., Polat, F., Alhajj, R.: Multi-objective optimization based location and social network aware recommendation. International Conference on Collaborative Computing: Networking, Applications and Worksharing pp. 233–242 (2014)
21. Pedro, D., Pazzani, M.: On the optimality of the simple bayesian classifier under zero-one loss. Machine Learning 29(2), 103–130 (1997)
22. Provost, F.J., Fawcett, T., Kohavi, R.: The case against accuracy estimation for comparing induction algorithms. In Proceedings of ICML pp. 445–453 (1998)
23. Sadileka, A., Kautz, H., Bigham, J.P.: Finding your friends and following them to where you are. Proceedings of the fifth ACM international conference on Web search and data mining pp. 723–732 (2012)
24. Salton, G.: Introduction to modern information retrieval. McGraw-Hill (1983)
25. Scellato, S., Noulas, A., Mascolo, C.: Exploiting place features in link prediction on location-based social networks. Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining pp. 1046–1054 (2011)
26. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Advances in Artificial Intelligence pp. 1–19 (2009)
27. Su, X.Y., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Advances in Artificial Intelligence archive 2009(4), 1–19 (2009)
28. Tang, J., Hu, X., Liu, H.: Social recommendation: a review. Social Network Analysis and Mining 3(4), 1113–1133 (2013)
29. Taskar, B., Wong, M.F., Abbeel, P., Koller, D.: Link prediction in relational data. In NIPS pp. 1–9 (2003)
30. Trattner, C., Steurer, M.: Detecting partnership in location-based and online social networks. Social Network Analysis and Mining 5(1), 1–15 (2015)
31. Vapnik, V.: The nature of statistical learning. Springer-Verlag (1995)
32. Veloso, M., Jorge, A., P, J, A.: Model-based collaborative filtering for team building. Proc. of ICEIS pp. 241–248 (2004)
33. Xiao, X., Zheng, Y., Luo, Q., Xie, X.: Inferring social ties between users with human location history. Journal of Ambient Intelligence and Humanized Computing 5(1), 3–19 (2014)
34. Yu, X., Pan, A., Tang, L.A., Li, Z., Han, J.: Geo-friends recommendation in gps-based cyber-physical social network. International Conference on Advances in Social Networks Analysis and Mining (ASONAM) pp. 361–368 (2011)
35. Zheng, V.W., Zheng, Y., Xie, X., Yang, Q.: Collaborative location and activity recommendations with gps history data. World Wide Web Conference Series pp. 1029–1038 (2010)
36. Zheng, Y., Zhang, L., Ma, Z., Xie, X., Ma, W.: Recommending friends and locations based on individual location history. ACM Trans. Web 5(1), 1–44 (2011)
37. Zhou, D., Wang, B., Rahimi, S.M., Wang, X.: A study of recommending locations on location-based social network by collaborative filtering. The 25th Canadian Conference on Artificial Intelligence pp. 255–266 (2012)
38. Zhu, J., Xie, Q., Chin, E.J.: A hybrid time-series link prediction framework for large social network. DEXA pp. 345–359 (2012)