Where Will You Tweet Tomorrow?
---Geo-based Social Analysis of Twitter

Ran Liu 15307110414, Jiwen Zhang 16307110435, Yin Wan 16307110431

Abstract

Predicting interactions between objects in a heterogeneous information network is a significant task in social network analysis. Recently, the rapid development of social networking platforms and the pervasive use of mobile phones have greatly promoted the study of location-based social analysis. In this essay, we study the heterogeneous network which is composed of locations and social networks of users extracted from Twitter. After calculating basic properties of our dataset, we detect communities, build recommendation systems, and study the trajectory of a representative user in this paper.

Keywords: Twitter, Community Detection, Recommendation System, Trajectory, Geo-location

1. Introduction

Location-based community detection and link prediction have drawn growing attention due to their practical significance and application potential. Most of the existing research exploring geo-location-based link prediction systems investigate two kinds of geo-location-based social networks. The first kind is human mobility patterns research, which requires large-scale data collection to investigate human dynamic traces. For the last decade, mobility-related problems have been widely studied due to the pervasive use of smartphones. The second kind of location-based social network research depends on a completely different type of dataset, which is the "check-in" dataset. While the dataset of mobility study provides scholars continuous and complete trajectory information, "check-in" dataset contains only certain location information at a certain time. Although the sparsity of dataset cause difficulty in investigating, solving such kind of problem may provide fundamental breakthrough and insights to heterogeneous information graph research.

To investigate such kind of problem, we selected social networking platform Twitter and the location information of tweets as our target data source. By mining data from Twitter, we acquired massive data about locations, users, their social networks, and their interactions. Related properties of our dataset were calculated and were interpreted from a sociology perspective. With this dataset, we selected some classic algorithms to detect communities and built appropriate recommendation systems. Finally, we studied the trajectory of a representative user and attempted to recommend new locations based on our existing information.

This essay is organized as follows: in Section 2, we describe the collection and cleansing procedures of our dataset and introduce some basic properties of the dataset. Section 3 describes location-based community detection based on fast-unfolding. Next, in Section 4 we build recommendation systems. In Section 5 we select a representative user to see its trajectory and qualitatively study its relationship with the social network and we apply recommendation system to this user. Finally, we summarize our findings and conclude with Section 6.

2. Dataset

2.1 Dataset collection

Being one of the most popular social networking services, Twitter is a platform where users post and communicate via short messages known as "tweets". Twitter allows users to follow each other and the online friendships created a directed social network of users. Also, Twitter allows users to post tweets with self-defined or auto-defined geo-tags which may provide additional location information of users. Geo-tags and users who provide them created a heterogeneous information graph containing location information.

In order to acquire a social relationship and location information, we constructed two datasets as follows:

user dataset: {user id, screen name, user location, user created at, friends}

tweet dataset: {user id, tweet text, tweet location, tweet bounding box, tweet created at}

where *user dataset* is dataset obtained from user information and *tweet dataset* is dataset obtained from tweet information. Two datasets can be connected via *user_id*. In *user dataset*, *user location* is self-defined location filled when creating user profile, *friends* is users followed by this user, and *user created at* is the time when user created his profile. In *tweet dataset*, *tweet text* is the contents of tweet, *tweet location* containing country code of tweet, self-defined or auto defined geo-location of tweet, and *tweet bounding box* which is the approximate latitude and longitude of tweet, and *tweet created at* is the time when user created this tweet.

Twitter is committed to protecting the privacy of users and thus Twitter users are free to decide what information will be open to public when posting their tweets. Therefore, although there are numerous twitter users and tweets, it is still not easy to scrape complete data of Twitter users and tweets, let alone tweets with geo-tag. Thus, the following procedures are designed to scrape data for our geo-based social analysis of Twitter.

Thanks to Stream provided by Twitter, 1394 seed users are saved by scraping users who posting Tweets with geo-tag real time. Then we find the friends of seed users and collect 3570724 users in total. From these users, we scrape the tweets with geo-tag from their timelines and get 289917 tweets. Then we filter the users whose accounts are protected or whose tweets without geo-tag and prune their *friends* columns to get a *user dataset* with 4918 users.

In order to get valid data, especially valid location information (what we consider is merely the information of cities and states in US), we cleanse our data using Google Map service. Firstly, non-US tweets are deleted according to their country code and 235259 tweets remain. Subsequently, we identified nonstandard tweet locations (e.g. self-defined locations) and return tweets' city and tweets' state by searching their latitude and longitude in Google Maps. Finally, after data cleansing, we get a *user dataset* of 4508 users and a *tweet dataset* of 209736 tweets. Also, our *tweet dataset* covers location information of 3878 American cities.

2.2. Dataset properties

After obtaining our dataset, we briefly present and describe some basic properties of our dataset. As for the social network constructed via the *user dataset*, the degree distribution is calculated as shown in fig 1. The degree distribution does not strictly follow the power law distribution but is more in line with the actual network. Since the figure clearly shows a tail effect: the number of people with a smaller degree is the majority.

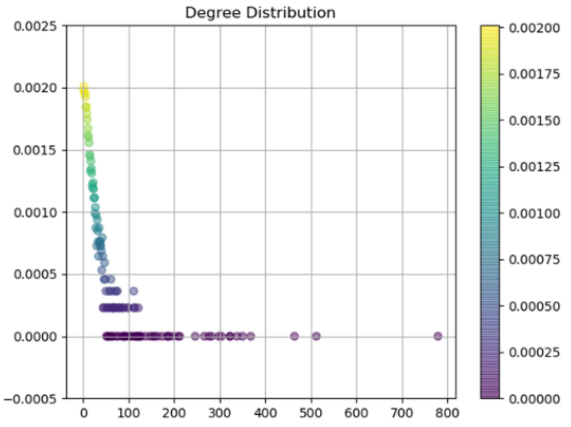


Fig.1. Degree distribution

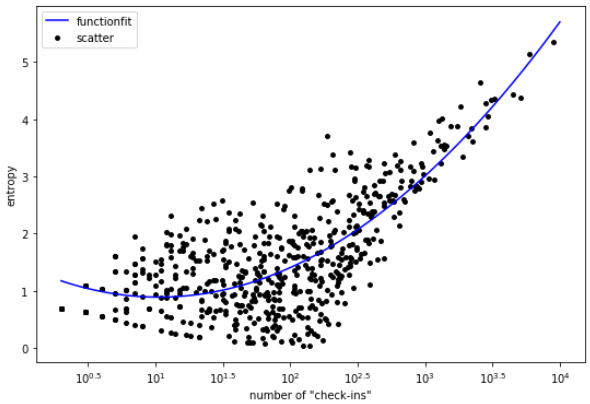


Fig.2. Entropy of place k

As for the heterogeneous graph that is composed of location and user information, we defined entropy and possibility of friendship to measure the properties of each location. Since we want to evaluate the importance of locations and their contribution to social relationships, we define entropy for each location. In ecology, researchers use place entropy to measure place biodiversity: diverse environment tend to exhibit uniform distribution of species (users in our case) while less diverse environment tend to exhibit skewed distribution. Entropy can be defined as follows:

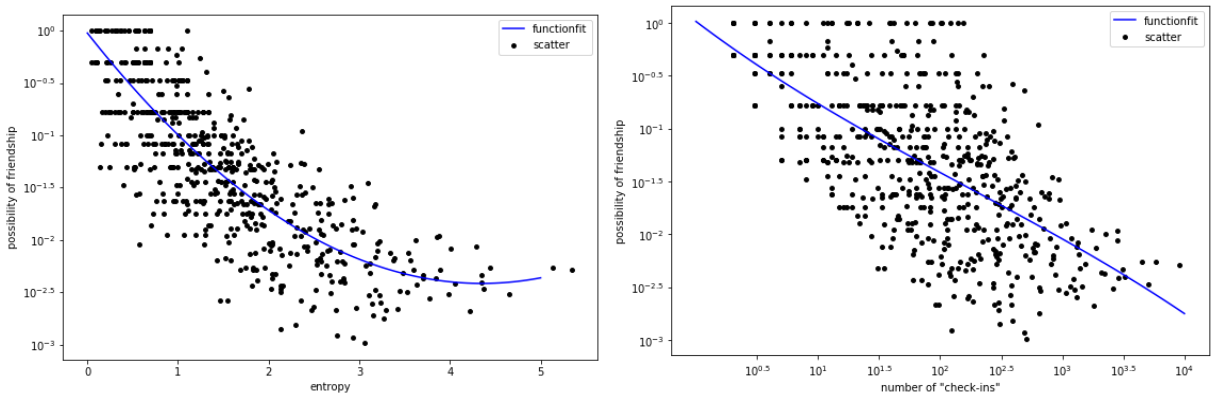
$$E_k = -\sum_{u_i \in \Phi_k} q_{ik} log q_{ik}$$

where Φ_k is the set of all users who have been in place k, and q_{ik} is the fraction of tweets that user i has at location k. Entropy of our locations range from 5 to 0. Generally speaking, locations with more tweets have higher entropy and locations with ~100 tweets have highest range of entropy. Also, it can be proven that entropy is an excellent index to measure the popularity of locations. Top five popular locations ranked by entropy from our dataset are New York city in New York, Los Angeles in California, Eden in Texas, Houston in Texas, and Chicago in Illinois. These five cities have 80% similarity with five cities ranked as the most popular travelling places in the US.

Also, we define the friendship possibility of location k, which is the total friendship pairs in all possible pairs:

$$p_k = \frac{f_k}{|\Phi_k| \times (|\Phi_k| - 1)}$$

where Φ_k is the set of all users who have been in place k, and f_k is the total friendship relationships at location k.



(a) Entropy (b) Number of check-ins

Fig.3. Possibility of friendship at place k

The relationship between friendship possibility and place entropy, as well as the relationship between friendship possibility and total tweets in certain places are described in fig 3(a) and fig 3(b). As the curve fits, places with higher entropy not only have more tweets but also have lower friendship possibility. This phenomenon can be explained by two assumptions. On the one hand, we can assume that place entropy can be used to evaluate if a place is suitable for traveling. Large cities have travelers from different places with comparably sparse friendship while small cities tend to attract travelers who share information. On the other hand, we can assume that people live in small cities or have similar taste in the selection of traveling places tend to become friends. Also, the total number of tweets can also reflect this result but entropy is a better measurement property.

3. Location-based Community Detection

In this section, we detect communities from our dataset, which contains 4508 users, 3878 cities, and 209736 interactions. By detecting communities, we are able to identify groups of users and locations that are closely related to each other. Hence, we can better understand the underlying phenomena and better exploit them. In our case, users and locations can be separated into different communities according to their interactions. We intended to detect location-based communities employing some classic algorithm and exploit its information.

In general, we propose two basic assumptions. Firstly, users are interested in locations where they have tweeted, which means each location reflect part of the preference of the user. Secondly, users' social connections have an influence on their locations, which means people tend to go where their friends have been to. Based on these two assumptions, we propose the following community detection method in our heterogeneous information graph.

Our community detection is based on fast-unfolding algorithm. Fast-unfolding is a heuristic method that is based on modularity optimization. The modularity of a partition is a number that measures the density of links inside communities as compared to links between communities. In the case of weighted networks, it is defined as:

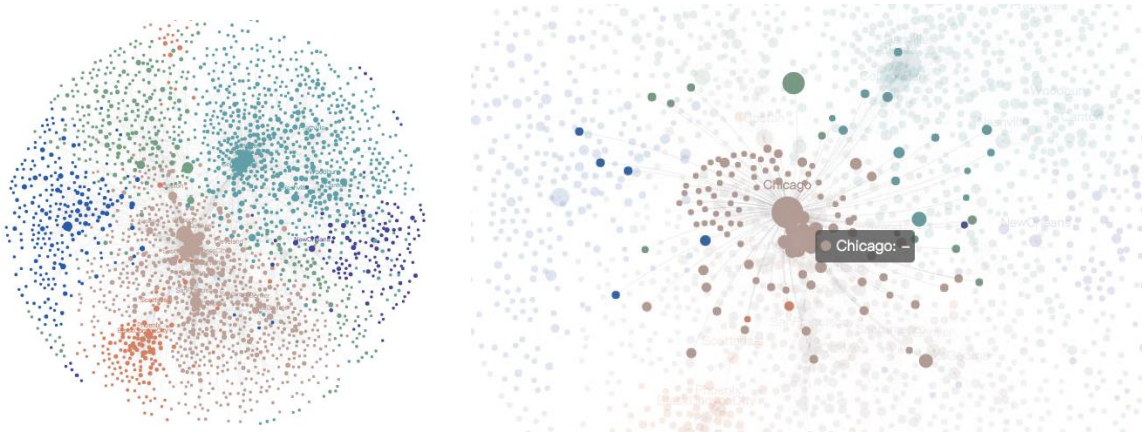
$$Q = \frac{1}{2m}\sum_{i,j}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j)$$

where A_{ij} is the weight of edge (i,j), k_i is the degree of vertex i, c_i is the community of vertex i, and m is the total degree of the network. Modularity optimization and community aggregation are the two core procedures of fast-unfolding. Modularity optimization is the procedure of dividing vertices to increase the value of modularity. Community aggregation is the procedure of building new graphs by aggregating communities to vertices. By repeating such procedures, we can get a final graph.

However, our existing dataset can only provide us with an unweighted network while community detection would be unreliable based on parameter of duality. That is to say that our dataset (whether if a location has been visited

before) cannot present the actual interest of users via fast-unfolding. Hence, we use deepwalk algorithm to evaluate the importance of locations. Our strategy is that: for every location, we start random walk of length six for 20 times. Our selected length can cover all of the network since reality social networks usually have average length of 4 to 5. Also, the random walk we started is a biased random walk since our start-points are locations. According to this strategy, we can get the distance of two nodes according to our embedding, and thus we can build a weighted heterogeneous information graph. We applied fast-unfolding algorithm to detect community and detected 22 communities from 4508 users and 3878 cities. Our result can be visualized as follows.

The following picture is a local visualization of Chicago. Being a large city and one of the central city of the US, many users who have been in Chicago are included as a same community. San Francisco is also in this community since the social network bonding of *user dataset* who have been in San Francisco and Chicago are close. Hence, this local structure is an example of locations being dragged to one community.



(a) Overall community detection map (b) Partial community detection map – Chicago

Fig.4. Community detection

4. Recommendation System

Note that 3878 US cities are well distributed in 51 US states, which means that geo-data is relatively complete. However, the interactions between users and geo-locations are quite sparse since 209736 interactions are only 1.2% of full interactions. Therefore, there is a cold start problem.

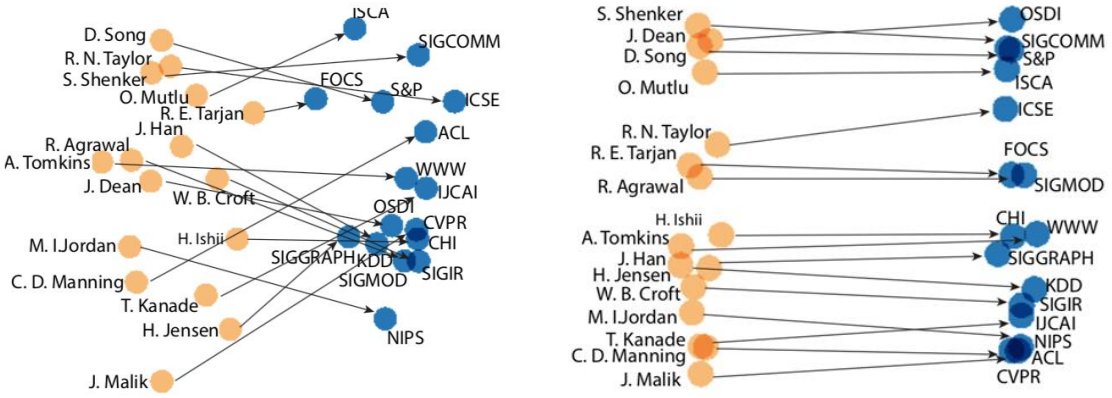
4.1. Data Pre-Processing

Following things have been done in order to conquer the cold start problem and make data easy to use in the recommendation system.

After generating a continuous unique ID for both users and cities using Python, we perform sentiment analysis to tweets. Geo-locations are extracted from users' tweets. Thus 209736 tweets are another important information that can be included in the recommendation system. It is worth mentioning that the content of tweets is not important. What is important is user's emotion reflected by tweets, which can partially represent the user's preferences towards the location.

Therefore, pre-trained (by microsoft) sentiment analysis model is employed to extract emotions from users' tweets. Model returns a score between 0-1, where 0 is negative emotion and 1 is positive emotion.

Metapath2Vec is introduced to deal with the cold start problem, thus we use it to do node embedding. Note that Metapath2Vec can not only outperform most embedding models in various heterogeneous information network but also discern the structural and semantic correlations between diverse network objects. This two main feature of Metapath2Vec ensures its effectiveness in this task.



(a) DeepWalk / node2vec (b) Mepath2vec++

Fig.5. 2D PCA projections of the 128D embedding

To mine the relationship of user-friends and user-friends-cities, initially the metapath scheme is

$$P : \text{V} \rightarrow \text{I} \rightarrow \text{I} \rightarrow \text{V} \rightarrow \text{I} \ldots$$

However, metapath start from V would lead to incomplete user traversal. Also, this metapath may again and again backtrack to the start node if it is a hot tweet place. Therefore, after several trails the matapath scheme was modified as

$$P : \text{I} \rightarrow \text{V} \rightarrow \text{A} \rightarrow \text{V} \rightarrow \text{I} \rightarrow \text{I} \rightarrow \text{V} \ldots$$

Firstly, starting from user makes sure all user will be travelled completely. Secondly, an added state node A will expose more location diversity during transitions.

In short, each user node and city node was embedded as a 128-dimensional vector.

4.2. Model Selection

Two recommendation models were selected among extensive models.

(a) Neural Collaborative Filtering -- NCF

NCF is an extension of collaborative filtering. The key idea of collaborative filtering is to model the interaction between user and item features. The improvement made by NCF is that it replaces the inner product with a neural architecture that can learn an arbitrary function from data.

$$\hat{y}_{ui} = f(P^T v_u^U, Q^T v_i^I | P, Q, \Theta_f)$$

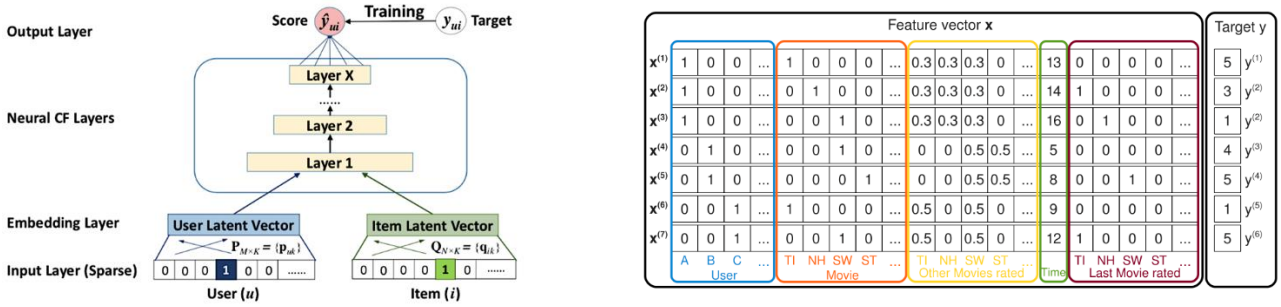
Where $P \in \mathbb{R}^{M \times K}$ and $Q \in \mathbb{R}^{N \times K}$ denote the latent fator matrix for users and items; and Θ_f denotes the model parameters of interactions function f. Since the function f is defined as a multi-layer neural network, it can be formulated as

$$f\left(P^T v_u^U, Q^T v_i^I\right) = \phi_{out}(\phi_X(\ldots \phi_2\left(\phi_1\left(P^T v_u^U, Q^T v_i^I\right)\right)\ldots))$$

By define the squared loss function as:

$$L_{sqr} = \sum_{u,i \in Y \cup Y^-} w_{ui}(y_{ui} - \hat{y}_{ui})^2$$

optimization can be done by performing stochastic gradient descent(SGD). The advantage of NCF is that it can effectively model implicit feedback when user satisfaction is not observed and there is a natural scarcity of negative feedback. Since our data only have user—geo interactions with no user ratings, NCF is highly in line with our recommendation task.



(a) Neural collaborative filtering (b) Factorization machine

Fig.6. Framework

(b) Factorization Machine – FM

FM is a classic model proposed by Rendle in 2010. FM can model all interactions between variables using factorized parameters.

$$\hat{y}(\boldsymbol{x}) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{i=i+1}^{n} < v_i, v_j > x_i x_j \qquad (*)$$

where the model parameters that have to be estimated are

$$w_0 \in \mathbb{R}, \qquad \boldsymbol{w} \in \mathbb{R}^n, \qquad V \in \mathbb{R}^{n \times k}$$

Note that a row vi within V describes the i-th variable with k factors where k is a hyperparameter that defines the dimensionality of the factorization. Where w_0 is the global bias, w_i models the strength of the i-th variable, and $w_{ij} \coloneqq < v_i, v_j >$ models the interaction between the i-th and j-th variable. Instead of using an own model parameter $w_{ij} \in \mathbb{R}$ for each interaction, the FM models the interaction by factorizing it.

Besides, by applying matrix calculation tricks, second order item in (*) can be computed as

$$\sum_{i=1}^{n} \sum_{i=i+1}^{n} < v_i, v_j > x_i x_j = \frac{1}{2} \sum_{f=1}^{k} \left(\left(\sum_{i=1}^{n} v_{i,f} x_i \right)^2 - \sum_{i=1}^{n} v_{i,f}^2 x_i^2 \right)$$

And time complexity of computing (*) can be reduced from $O(kn^2)$ to $O(kn)$. Therefore, FM allows high quality parameter estimates of higher-order interactions under sparsity, which is compatible with the data.

RMSE is chosen as the error estimator:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2}$$

since this is an unsupervised learning task and RMSE can intuitively reflect the quality of the model.

4.3. Feature Engineering

After data pre-processing, several useful attributes were extracted for model to use. However, a question is: attributes may contain some interference information and thus weaken the power of model. Therefore, permutations of these attributes were tested to find the best combination.

The outcome based on error estimator RMSE is shown here:

Structure	Contents	NCF	FM
1	User&geo one hot	0.3033	0.0716
2	User&geo one hot + Embeddings	0.2233	0.0392
3	User&geo one hot + Emotions	0.2933	0.0722
4	User&geo one hot + Embeddings + Emotions	0.2193	0.0623

Table.1. The outcomes of 4 different structures

It is not difficult to see that structure 2 in the above table has the best performance. Thus future analysis would focus on structure 2.

4.4. Parameter Adjustment

From results of feature engineering, we found that the RMSE of FM in trainset and testset were quite different from NCF.

	NCF		FM	
	trainset	testset	trainset	testset
Structure 2	0.2233	0.2141	0.0392	0.1542

Table.2. The results of NCF and FM

To find a set of parameters, which can reduce the difference between FM in training and test data sets as well as compare the performance of different parameters on model, parameters of the two models were adjusted according to exponential increase.

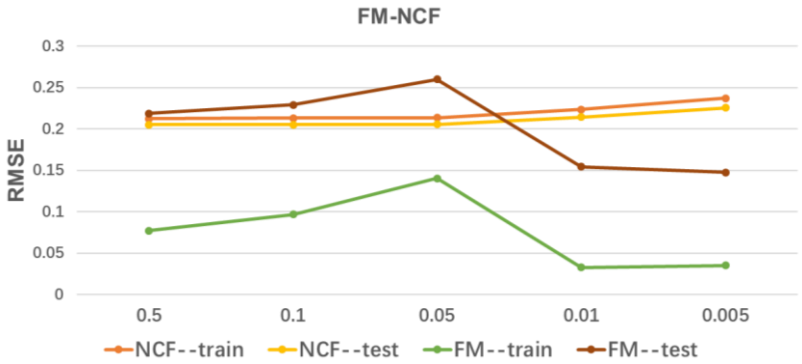


Fig.7. Parameter adjustment line graph

We have mainly adjusted learning rate. From the graph, we can find that FM is better than NCF in training set. FM is superior to NCF in testset only when the learning rate is small.

In short, NCF is a relatively stable model: learning rate will not have a great impact on the results. It is more robust, more suitable for handling some large-scale tasks. While FM has a strong fitting ability, which may lead to over-fitting when lr is relatively large. However, when the lr is small, the model can achieve better results for a specific problem.

5. Trajectory

Apparently, tweets' geo-tags vary in different times. Therefore, we can derive trajectories of a user if connecting different geo-locations in order of his/her timeline. In this section, we selected a representative user 104249727 to investigate its trajectory and its relationship with its social network.

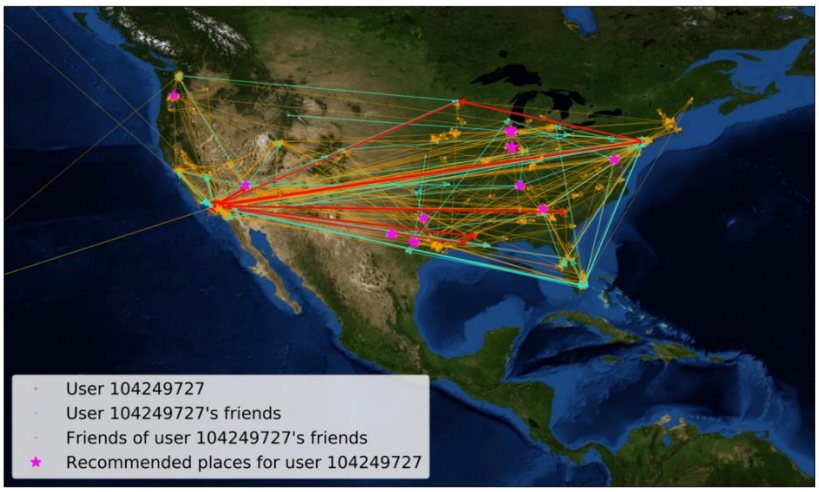


Fig.8. Trajectories and recommend places on US map

Red arrows depicts the trajectories of user 104249727 while the cyan is his/her friends and the orange is the friends of his/her friends. Some arrows with different colors are clearly coincident on the map. Hence, it is safe to say that users with social relationship are prone to have some same or similar trajectories on Twitter.

Combined with the recommendation system in Section 4, top 10 recommended places for user 104249727 are also marked purple on the map. As we can see from the figure above, recommended places are exactly located at the intersections of trajectories of his/her friends or the friends of his/her friends. This means the frequent places of friends are potential to be recommended to the user in our recommend system.

6. Conclusion

In this report, we finish data mining of 209736 Tweets with geo-tag and detect 22 communities from 4508 users and 3878 cities. Additionally, we recommend new places for users and analyze the trajectories of a representative user as well as his/her friends and the friends of his/her friends. Besides, we finish link prediction of location.

Our future work includes further exploitation and analysis of our dataset. For example, 1) data cleansing and analysis of some other columns in our dataset, such as *user location* or *user created at*; 2) natural language processing of *tweet text*; 3) recommendation based on social network combined with *user location* and *tweet text*.

References

[1] Cao, G. , Wang, S. , Hwang, M. , Padmanabhan, A. , Zhang, Z. , & Soltani, K. . (2015). A scalable framework for spatiotemporal analysis of location-based social media data. Computers, Environment and Urban Systems, 51, 70-82.

[2] Kim, S. , Jeong, S. , Woo, I. , Jang, Y. , Maciejewski, R. , & Ebert, D. . (2017). Data flow analysis and visualization for spatiotemporal statistical data without trajectory information. IEEE Transactions on Visualization and Computer Graphics, 1-1.

[3] Yang, D. , Zhang, D. , Zheng, V. W. , & Yu, Z. . (2015). Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 45(1), 129-142.

[4] Scellato, S. , Noulas, A. , & Mascolo, C. . (2011). Exploiting Place Features in Link Prediction on Location-based Social Networks. Acm Sigkdd International Conference on Knowledge Discovery & Data Mining. ACM.

[5] Cranshaw, J. , Toch, E. , Hong, J. , Kittur, A. , & Sadeh, N. . (2010). Bridging the gap between physical location and online social networks. Acm International Conference on Ubiquitous Computing. ACM.

[6] Blondel, V. D. , Guillaume, J. L. , Lambiotte, R. , & Lefebvre, E. . (2008). Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(10), 0-0.

[7] Perozzi, B. , Al-Rfou, R. , & Skiena, S. . (2014). Deepwalk: online learning of social representations.

[8] Dong, Y. , Chawla, N. V. , Swami, A. , Dong, Y. , Chawla, N. V. , & Swami, A. . (2017). metapath2vec: Scalable Representation Learning for Heterogeneous Networks. Acm Sigkdd International Conference on Knowledge Discovery & Data Mining. ACM.

[9] Shi, C. , Zhang, Z. , Luo, P. , Yu, P. S. , Yue, Y. , & Wu, B. . (2015). Semantic Path based Personalized Recommendation on Weighted Heterogeneous Information Networks. the 24th ACM International. ACM.

[10] Rendle, S. . (2011). Factorization Machines. IEEE International Conference on Data Mining. IEEE.

[11] Zhao, H. , Yao, Q. , Li, J. , Song, Y. , & Lee, D. L. . (2017). Meta-Graph Based Recommendation Fusion over Heterogeneous Information Networks. Acm Sigkdd International Conference on Knowledge Discovery & Data Mining. ACM.

[12] He, X. , Liao, L. , Zhang, H. , Nie, L. , Hu, X. , & Chua, T. S. . (2017). [acm press the 26th international conference - perth, australia (2017.04.03-2017.04.07)] proceedings of the 26th international conference on world wide web, - www \"17 - neural collaborative filtering. 173-182

Roles and Responsibilities of the Team

Ran Liu: Data cleansing and analysis & Trajectory
Jiwen Zhang: Location-based community detection & Recommendation system
Yin Wan: Data scraping & Trajectory