SNA Final Project Report
Link Prediction on Location-Based Social Network

B01901130 邱德旺
B01902024 陳亮瑋
B01902067 賴至得

1. INTRODUCTION

Link prediction systems have been largely adopted to recommend new friend in social networks. With location-based data, it is more likely to take an advantage of an additional source of information. In this project, we are going to handle a link prediction problem with location-based social network to predict the potential relationship via check-in records. We describe a supervised learning framework with location-based features in this project, which can be taken as a binary classification problem.

First, we introduce the dataset we use in section 2. Some previous works have been done to exploit location-based features in section 3. In addition to those features, we extract some other useful features in this project. We introduce our model structure and the features in detail in section 4. The features can be separated into 3 categories-- social features, location-based features, and heterogeneous-graph features. The experiment results are shown in section 5. Based on those experiments, we discuss the results and make conclusions in section 6.

2. DATASET

2.1. Gowalla

Gowalla is a location-based social networking service. It provides a platform for users to check in spots and share it with their friends. In this project, we use the Gowalla dataset provided by TA. We have the information of each user and each spot, the friend relationship of users, and check-in records between 2009 March and 2011 December. The format of check-in records is given below:

[user]	[check-in time]	[latitude]	[longitude]	[location id]
196514	2010-07-24T13:45:06Z	53.3648119	-2.2723465833	145064
196514	2010-07-24T13:44:58Z	53.360511233	-2.276369017	1275991
196514	2010-07-24T13:44:46Z	53.3653895945	-2.2754087046	376497
196514	2010-07-24T13:44:38Z	53.3663709833	-2.2700764333	98503
196514	2010-07-24T13:44:26Z	53.3674087524	-2.2783813477	1043431
196514	2010-07-24T13:44:08Z	53.3675663377	-2.278631763	881734
196514	2010-07-24T13:43:18Z	53.3679640626	-2.2792943689	207763
196514	2010-07-24T13:41:10Z	53.364905	-2.270824	1042822

2.2. Input and Output

We have 5 data files from the Gowalla dataset as our inputs:

(1) users_info_new.dat – contains ID of each user, their hometowns, and their follower counts

(2) spots_info.dat – contains ID of each spot, their categories, their latitude and longitude

(3) checkins_info.dat – each line contains a ID of an user and a sequence of its check-in records. Each check-in record is a tuple: (check-in time, check-in spot ID)
(4) gowalla.train.txt – The original user-user friendship data is divided into training and testing data. This training file contains 191995 user-user pairs, 40% of them have edges in the original friendship.
(5) gowalla.test.txt –The original user-user friendship data is divided into training and testing data. This testing file contains 86986 user-user pairs, 40% of them have edges in the original friendship.

We use (1)~(4) to extract features and train our model, and predict whether the user pairs in (5) really have edges.

3. RELATED WORK

The link prediction problem in social networks has been under research for many years. Early works often adopt unsupervised approaches. More recently, supervised approach has been shown effective if we limit the candidates to two-level friends-of-friend. In location-based social network (LBSN), Cho et al. analyzed LBSNs and find that human movement is geographically limited.[1] Scellato et al. used check-in data to generate features for supervised learning, such as common check-in counts, geo-distance, or entropy features.[2] Mengshoel et al. found that feature selections and filtering other data sets have a major impact on the accuracy of link prediction in LBSNs.[3] Bayrak et al. combined the previous works, recommended 5 new features, and apply multiple ML algorithms to them,[4] which has been the state-of-the-art.

4. MODEL

4.1Model Structure Introduction

We are going to construct a supervised machine learning model. First, with the input mentioned above, we generate three kinds of features:
(1)Social/homogeneous graph features, which are based on the user-user friendship graph. Well-known link prediction features such as Jaccard coefficient and Adamic-Adar scores are in this category.

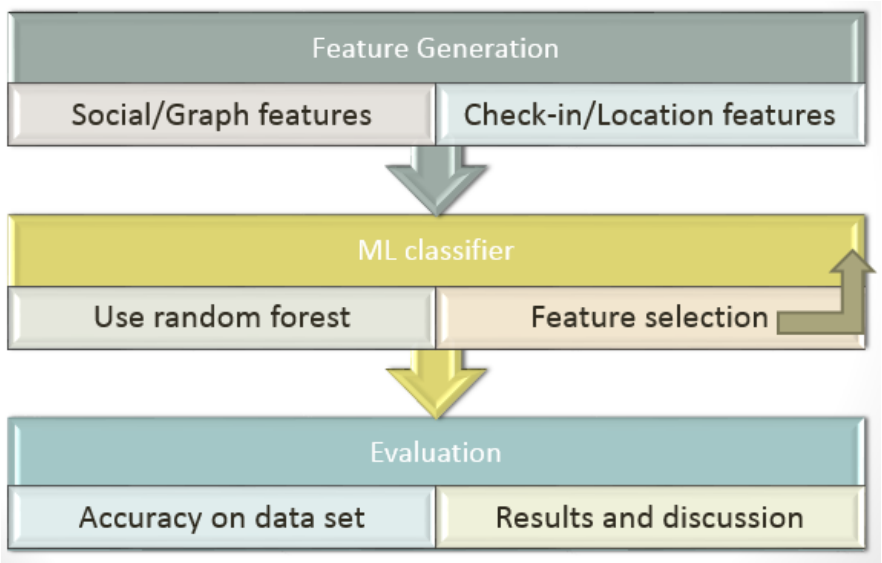
---

[1] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks.In Proc. of KDD-11, pages 1082–1090, 2011.
[2] S. Scellato, A. Noulas, and C. Mascolo. Exploiting place features in link prediction on location-based social networks. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11, pages 1046–1054, New York, NY, USA, 2011. ACM
[3] O. J. Mengshoel, R. Desai, A. Chen, and B. Tran. Will we connect again? machine learning for link prediction in mobile social networks. In Eleventh Workshop onMining and Learning with Graphs, MLG 2013, 2013.
[4]Ahmet Engin Bayrak and Faruk Polat. Contextual Feature Analysis to Improve Link Prediction for Location Based Social Networks.SNAKDD'14 Proceedings of the 8th Workshop on Social Network Mining and Analysis

(2)Check-in/location features, which based on the check-in records and the geographical information from user and spot profile. Lots of them have been shown effective in previous LBSN works.

(3)**Heterogeneous graph features**. Except for the user-user friendship graph, there are three kinds of different graphs constructed by the information we know. We propose several new features based on those graphs.

After generating those features for each user-user pair instance, we put them into our ML classifier as a binary classification problem. We use random forest here, for its convenience to perform feature selection via cross-validation and it's appropriate to handle category features. Finally, we output the prediction for each instance in our testing set.



4.2.Prediction Features

4.2.1 Social/homogeneous graph features

The followings are some topological features for an user pair (x, y) in the user-user friendship graph. $\Gamma(x)$ denotes the friends of user x.

Index	Feature	Definition
g0	Adamic-Adar Score	$\sum_{z \in \Gamma(x) \cap \Gamma(y)} 1/log\lvert\Gamma(z)\rvert$
g1	Jaccard Coefficient	$\lvert(\Gamma(x) \cap \Gamma(y))\rvert \,/\, \lvert(\Gamma(x) \cup \Gamma(y))\rvert$
g2	Resource Allocation Index	$\sum_{z \in \Gamma(x) \cap \Gamma(y)} 1/\lvert\Gamma(z)\rvert$
g3	Preferential Attachment	$\lvert\Gamma(x)\rvert \times \lvert\Gamma(y)\rvert$

g4	Shortest Path Length	The length of the shortest path from x to y. (If two nodes has an edge on the training graph G, we neglect the edge and then do the calculation)
g5	Common Neighbors	$\mid \Gamma(x) \cap \Gamma(y) \mid$
g6	Approximate Katz	$\bigcup_{u \in \Gamma(x),\, v \in \Gamma(y)} \{ (u,\, v) \mid u = v \text{ or } u, v \text{ connect} \} \mid$
g7	Follower Count Product	|Follower(x)|*|Follower(y)| (Follower number is obtained from the user profile)

4.2.2 Check-in/Location Features

The followings are some check-in/location features for an user pair (x, y).

Index	Feature	Definition
c0	Check-in Trip Length Sum	The sum of distance from first check-in location going through all check-in location to last check-in location of x and y.
c1	Same Day Common Location Count[5]	The number of two user x and y's common check-in records in the same day.
c2	Same Day Distinct Common Location Count	The number of two user x and y's common check-in records in the same day. The duplicate location will only be calculated once.
c3	Distinct Common Location Count	The number of two user x and y's common check-in records (not necessarily in the same day). Duplicate locations will only be calculated once.
c4	Same Day Common Location Count with Min Entropy	The number of two user x and y's common check-in records in the same day weighted by entropy Ek. Where qik is the fraction of check-ins of this spot of user_i on location k. $E_k = -\sum_{u_i \in \Phi_k} q_{ik} \log q_{ik}$
c5	Distance of Hometown	The distance of hometown of two user x and y.

[5] (Feature c1~c7 are from Exploiting Place Features in Link Prediction on Location-based Social Networks. ACM KDD 2011)

c6	Same Day Common Location Count with Distance from Hometown	The number of two user x and y's common check-in records in the same day weighted by the distance from hometown.
c7	Check-in Spot Product	The product between the number of spots that user x and user y have visited.
c8	Total Common Criends Closeness(TCFC) [6]	Where CFC(x, y) denotes common friends count of user x and user y. $$TCFC(x,y) = \sum_{z \in (\alpha_x \bigcup \alpha_y)} CFCx, z * CFC(y,z)$$
c9	Total Common Friends Check-in Count(TCFCC)	Where CCC(x, y) is the Common Check-in Count in the same day of user x and user y. $$TCFCC(x,y) = \sum_{z \in (\alpha_x \bigcup \alpha_y)} CCCx, z * CCC(y,z)$$
c10	Common Category Check-in Counts Product (CCCP)	Where ϕ(x,m) is the check-in count on category m of user x. $$CCCP(x,y) = \sum_{m \in (\omega_x \bigcup \omega_y)} \lvert\phi_{x,m}\rvert * \lvert\phi_{y,m}\rvert$$
c11	Common Category Check-in Counts Product Ratio(CCCPR)	$$CCCPR(x,y) = \frac{CCCP(x,y)}{\sqrt{\sum_{m \in \omega_x} \lvert\phi_{x,m}\rvert^2 * \sum_{m \in \omega_y} \lvert\phi_{y,m}\rvert^2}}$$

Note that though we have the hometowns from the user profile, the hometowns aren't formalized and they often represent a big city with radius more than 50 km!! In addition, nearly 50% of them are missing, so we cannot use them directly.

The original author of the paper solves this problem by take the most common check-in spots as the hometown. However, we observe users seldom check in the same spot, so it's hard to decide its hometown. We propose another method to calculate the latitude and longitude of the hometown. It's believed that human movement is geographically limited, and the check-ins may be around their home. Thus, we averages all the check-in records of a user to be its geographical hometown.

[6] (Feature c8~c11 are from Contextual Feature Analysis to Improve Link Prediction for Location Based Social Networks, SNA-KDD 2014)

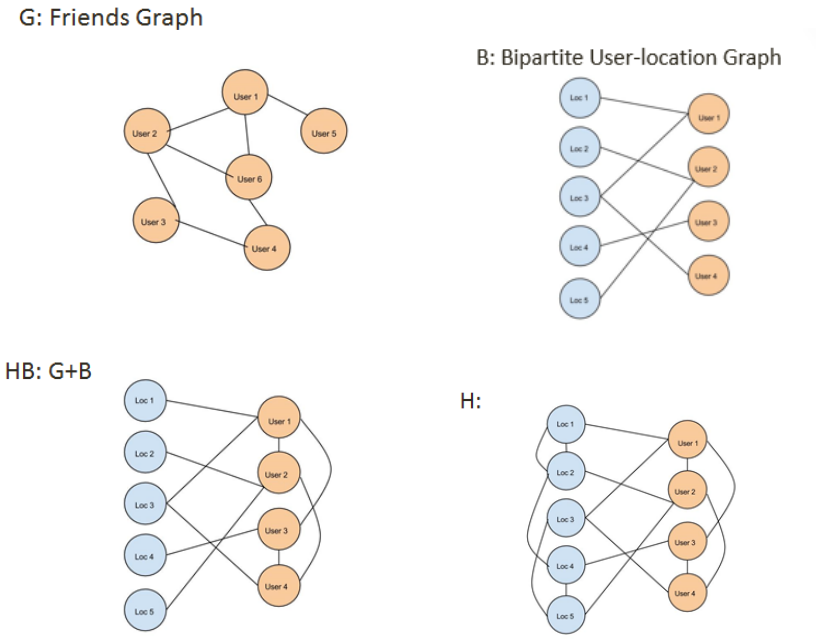
4.2.3 Heter-Graph Features

Except for the user-user friendship graph, we introduce three more graphs to generate features:

(1)Bipartite User-location Graph (B): Based on the check-in records, we construct this bipartite graph to model the check-in behaviors of users to locations without user-user or location-location relations.

(2)Half Bipartite User-location and User-user Graph (HB): We combine the original user-user graph G and the bipartite graph above to construct this graph HB. This graph aims to simultaneously model social relation and check-in behaviors.

(3)Full heterogeneous Graph(H): There should be location-location relations that has influence on the user check-in behaviors or even their friendship relations. Based on HB, we add edges between two locations. An edge is added if the Euclidean distance of the latitude and longitude of two locations is smaller than 0.1 (around 10 km in realistic world).



The followings are some features for a user pair (x, y) in heterogeneous graphs B, HB and H.

Index	Feature	Definition
h0	Approximate Katz on B	Using Approximate Katz on the graph B $Approximate\ Katz(x,\ y)\ =\ \lvert \bigcup_{u\in\Gamma(x),\ v\in\Gamma(y)} \{(u,\ v)\ \lvert\ u\ =\ v\ or\ u,\ v\ connect\}\ \rvert$
h1	Approximate Katz on HB	Using Approximate Katz on the graph HB
h2	Approximate Katz on H	Using Approximate Katz on the graph H

h3	shortest path on B	Using the inverse of shortest path length on the graph B
h4	clustering coefficient on H	The product of the clustering coefficient scores. $clustering\ coefficient(x) = \left\| \bigcup_{u \in \Gamma(x),\ v \in \Gamma(x)} \{ (u,\ v) \mid u,\ v\ connect \} \right\|$
h5	User location friends' location friend number	-Get friends of x that are location nodes in H, L1、L2、L3…..Lx x:1~n -Add all number of friends from L1 to Ln. -Do 1~2 on y -Use product of the results as feature.
h6	location friends' degree sum	-Get x's neighbors that are locations (we call them the location friends of x) -Sum their degrees -Do 1~2 on y -Use product of the results as a feature.
h7	Approximate katz for social graph between location friends	Count approximate katz for social graph between x's location friends and y's location friends
h8	Adamic adar score on H	Count adamic adar score on graph H
h9	Resource_allocation on H	Count resource_allocation on graph H
h10	Shortest path length on H	Count shortest path length on graph H
h11	Common neighbors on H	Count common neighbors on graph H

5. EVALUATION

5.1 Evaluation Metrics

We take "accuracy" score on testing data separated by TA as our evaluation metrics. There are no common imbalanced problem because 40% of the user pair instances are positive and 60% of the user pair instances are negative. First, we use only Adamic-Adar score (g0) as our baseline because the previous paper [7] show that this feature performs well among the user-user homo-graph features. Second, we try to add check-in/location features in previous papers with our baseline in order to show that those location-based features really

[7] S. Scellato, A. Noulas, and C. Mascolo. Exploiting place features in link prediction on location-based social networks. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge

help in the prediction. Then, based on all the useful user-user features and location-based features, we are going to add our new heterogeneous graph features to see whether those features really work. In all experiments, we use random forest classifier as our machine learning training model because of its efficiency and additional function of feature selection to help us to understand the importance of each feature. The parameters are (*n_estimators=1000*, *max_depth=20, others are as default in the sklearn library*).[8]

5.2 Experiment Results

Baseline:(g0) Adamic-Adar score : 0.631289669764

(1)Common Location Count features(c1~c4)

Features	g0	g0+c1	g0+c2	g0+c3	g0+c4
Accuracy	0.6312896	0.6452833	0.6452488	0.6489743	0.6369124

(2)Check-in Spot Product(c7)

Features	g0	g0+c7
Accuracy	0.6312896	0.6294499

(3)Geographical check-in features--latitude and longitude(c0, c5, c6)

Features	g0	g0+c0	g0+c5	g0+c6
Accuracy	0.6312896	0.6418682	0.6466746	0.6452488

(4)Closeness and category features (c8, c9, c10, c11)

Features	g0	g0+c8	g0+c9	g0+c10	g0+c11
Accuracy	0.6312896	0.6368434	0.6329799	0.6308067	0.6319105

(5)Aggregation

According to the above experiments, we choose useful features and assemble them together.

Add useful location features:

Features	g0	g0+g2+g4+g5+g6 (user-user features only)	g0+g2+g4+g5+g6+c0+c1+c3+c5+c6+c7+c8+c9+c10+c11(Add location-based features)

[8] sklearn.ensemble.RandomForestClassifier, http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Accuracy	0.6312896	0.676570692	0.707547603

(6)Mixed Graph features(h0 ~ h11)

We call the best combination above “m1”, and use m1 as the baseline to select useful heter-graph features.

Features	m1	m1+h0	m1+h1	m1+h2	m1+h3	m1+h4	m1+h5
Accuracy	0.70754	0.70750	0.70770	0.70793	0.70859	0.71127	0.70835

Features	m1+h6	m1+h7	m1+h8	m1+h9	m1+h10	m1+h11
Accuracy	0.70976	0.70741	0.70690	0.70765	0.70759	0.70655

After all the experiments, we choose our final model with the following features:
g0+g2+g4+g5+g6+c0+c1+c3+c5+c6+c7+c8+c9+c10+c11+h2+h3+h4+h5+h6+h9+h10
Accuracy :0.723829454512

6.DISCUSSION

As we mentioned in section 5.1, Adamic-Adar score has been shown to have good results on this Gowalla dataset in previous works. We try to combine other common topological features on the user-user homo-graph, some are useful but some aren’t due to the overlapping topological characteristics of those features. We try each combinations and find the best of them (g0+g2+g4+g5+g6).

We emphasize on analyzing the location-based features. Common Location Count features play an important role in our model. The users checking in the same place in the same time are believed more likely to be friends. If two users check in the same place that few people check in, this place may be important. However, in our data, few users check in the same location for more than 2 times, feature “Same Day Distinct Common Location Count(c2)" is similar to "Same Day Common Location Count(c1)". "Same Day Common Location Count with Min Entropy(c4)” isn’t much useful due to this. We observe that most users have at most check in 20 places, and lots of them check in exactly 20 places. Thus, "Check-in Spot Product(c7)" may not work.

We propose another definition of hometown, and the geographical features based on latitude and longitude do help in our prediction. Most friends don’t live far from each other, so the feature “Distance of Hometown(c5)” is quite reasonable. If two people check in the same place that is far from their hometown, they are more likely to have important relation(c6). "Check-in Trip Length Sum(c0)" is a feature we propose to model the movement behavior of people. The person who usually travels a long distance compared to another person who

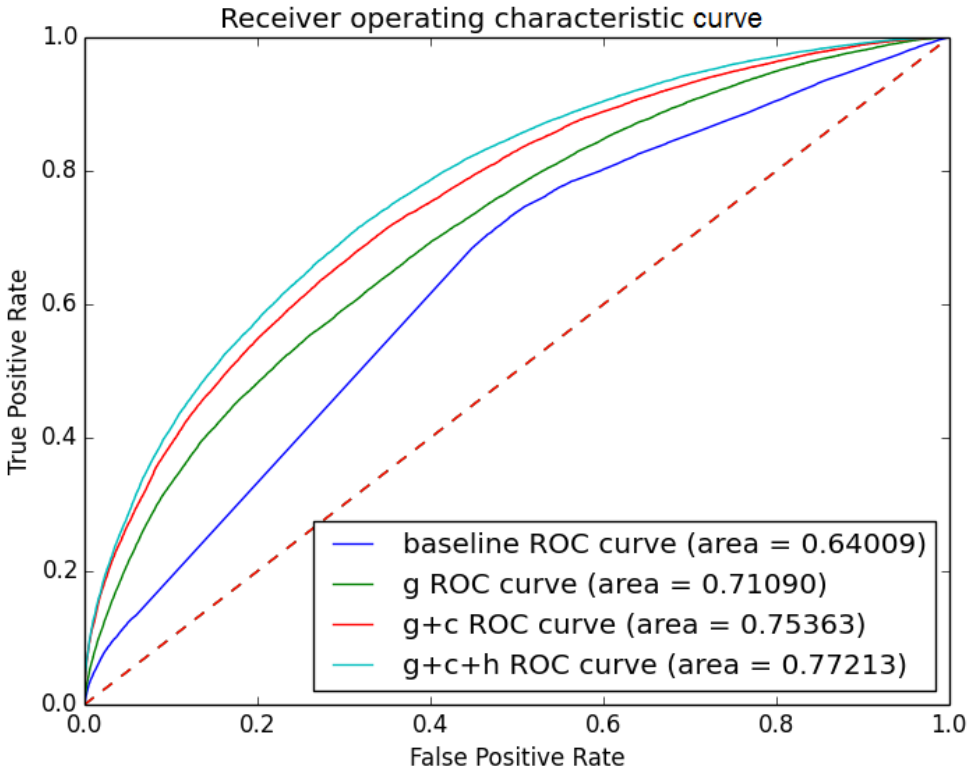
doesn't go far away may affect our model in a different way. Next, we apply the features (c8~c11) from the state-of-the-art[9]. But the features related to spot categories doesn't work.

Then, we try to model this problem with features in the heterogeneous graphs. We believe that the ensemble of all paths by Katz score may contain potential relations in those graphs, but it takes too much time to compute the real Katz score. We calculate the approximate Katz [10]instead, which is proved close to Katz score. It works on graph H and graph HB (feature h1, h2) but doesn't on B. The formula of the approximate Katz would only count the common neighbors(locations) due to the topological characteristics of a bipartite graph , and we have already calculate it in feature "Distinct Common Location Count(c3)". In addition, we think users that go to places which have higher centrality are more likely to be friends. Therefore, we use "Location Friends' Fegree Sum(h6)" as our feature and it proved to be useful. Then, we try some topological features on graph H. It turns out that some topological features work, some don't. The reason that those features that doesn't work may be that they are overlapping to features "Common Neighbors(g5)" and "Distinct Common Location Count(c3)". Therefore, "Common neighbors on H(h11)" doesn't work and so is "Adamic adar score on H(h8)". As for those features that work, they may show potential relations between users and locations, which do not represent by previous features we used. Thus, they do improve performance.

The following is the AUC-ROC curve on the testing set. The blue curve is for our baseline Adamic Adar score(g0), the green is for the user-user homo-graph features (g0+g2+g4+g5+g6) only, the red line is the resulted curve after we add location-based features, and the gray curve is for our final model. The curves below show that location-based features really help us to solve the link prediction problem in LBSNs, and our heterogeneous graphs modeling help to improve the performance.

[9] Ahmet Engin Bayrak and Faruk Polat. Contextual Feature Analysis to Improve Link Prediction for Location Based Social Networks.SNAKDD'14 Proceedings of the 8th Workshop on Social Network Mining and Analysis

[10] Computationally Efficient Link Prediction in a Variety of Social Networks,MICHAEL FIRE, LENA TENENBOIM-CHEKINA, RAMI PUZIS, OFRIT LESSER,LIOR ROKACH, and YUVAL ELOVICI, Ben-Gurion University of the Negev. ACM, 2013



Note that in the previous papers, the authors only choose two-layered friends-of-friends as potential candidates of friends. This evaluation method may results in very high AUC, but it's a biased sampling. In our experiments, we adapt all the features in the previous state-of-the-art. But instead, our candidates of friends aren't limited to friends-of-friend subgraph.