Social Network Analysis - Link Prediction

Chu-en Ho, *B00902006,* Hao-en Sung, *B00902064,* and Skyly, *R03922039*

***Abstract*—In this report, we first build location and graph features studied in related works, and test their performance. Then, some possibly effective features derived by us are put into later experiments. Finally, we conclude that location features do improve link predictions on social network.**

***Keywords*—Location-based Link Prediction, Social Network**

I. Introduction

With the exponential expansion of social network, it is more important to have a better insight on the relationship between people. To achieve higher accuracy rate on prediction of friend relationship in social network, many well-known methods and features have been studied. One relatively new idea to improve prediction preformace is to utilize the checkin data.

In this report, we first define our problem in section 2. In section 3, we derive some possible methodologies to extract features. The experiment design and experiment results are given in section 4 and section 5 separately. We will have our discussion and conclusion in section 6.

II. Problem definition

With the knowledge of partial social network and checkin data, the task is to predict the existence of friendship between two users. The evaluation loss function will be $0 - 1$ error.

To be clear, given $G = (N, E)$ where ther are N nodes, E edges in graph, and checkin data $C = (U, T, P)$ where user U checkin at location P on time tag T. For each query $Q = (U_1, U_2)$, the output should be 0 or 1 for the existence between U_1 and U_2.

III. Methodology

A. Location Stitching

There are 1280969 unique location IDs provided by *Gowalla*, but only 6442892 instances are in checkin data. The average number of checkin for each user is 5.029701, and the majority pairs of nodes never appear in same location. For Brightkite, there are 772966 unique location IDs and 4747281 insatnces in check-in data, and the average is 6.141642.

It is reasonable to doubt that some of the location IDs may target to a same location in real world. For example, users have checkins in the northern and southern part of Da-An Forest Park may use different location IDs. To solve this problem, we derived an easy-implemented algorithm that will gradually stitch two location IDs whose distance are smaller than r iteratively from longitude -180 to 180.

For *Gowalla*, if choosing r as 0.001, the number of unique location IDs reduces from 1280969 to 930288, however, it is hard for human to identify the differences in figures. The origin dot graph is showned in [Fig. 1].

For *Brightkite*, if choosing r as 0.001, the number of unique location IDs reduces from 772966 to 536052. The origin dot graph is shown in [Fig. 2].

The pseudo code for location stitching is attached in appendix.

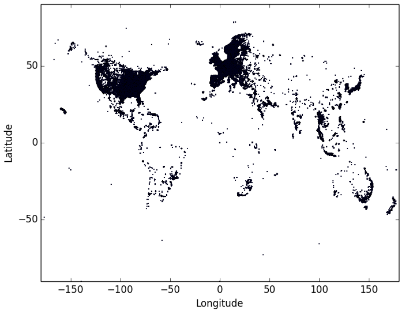

Fig. 1: Dot graph for Gowalla Origin Locations

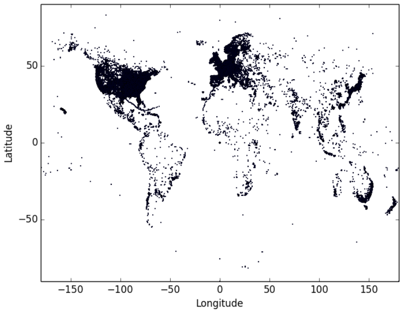

Fig. 2: Dot graph for Brightkite Origin Locations

B. Instance Sampling

Since the enormous missing features created with TA's sampled instances, we try to sample new instances from source data with equal postive and negative instances. More precisely, we define *qualified* pair that both users have recoreds in checkin data and they must appear in same location at least once. In this sampling setting, missing features problem is mostly solved. It helps us have a better insight into the importance of location-based features.

Last Edited 21:59 January 24, 2015.

1) Sampling on Orgin Figures: To sample positive instances, we find out all *qualified* edges in both *Gowalla* and *Brightkite* social networks, and there are 351452 and 141350 of them separately. We sample about 20 percent of such edges as positive instances, and leave the rest as shared graph.

To sample negative instances, we enumerate all *qualified* pairs that are not in social networks and their distance are exactly 2. There are 23497570 and 2599008 of them in both graph. We sample about 0.3 percent of such pairs as negative instances.

2) Sampling on Stitched Figures: There are about 367540 *qualified* edges in *Gowalla* and 155874 of them in *Brightkite*. We sample about 20 percent of such edges as positive instances, and leave the rest as shared graph.

There are 26562198 and 3446144 *qualified* pairs that are not in social networks and their distance are exactly 2. We sample 0.3 percent of such pairs as negative instances.

C. Feature Manufactory

We will first give a brief introduction about the features suggested by other papers. After that, we create several graph features from social network and further location features from checkin data.

1) Old Location Features: Most features are straight forward, explained by origin author: "The two features *common_p* and *overlap_p* denote respectively the number and the fraction of common places between two users, while *w_common_p* takes into account the number of check-ins of both users and *w_overlap_p* is given by the cosine similarity of the two check-in vectors. These features come from an observation: people tend to visit places where their friends had visited.

However, a place like a famous tourist attraction where most of the people will check-in only one time, suggest less importance. Hence, the paper proposed an entropy-based method. Let C_k^P be the total number of check-ins all users have at place m_k and $q_{ik} = c_{ik}/C_k^P$ the fraction of check-ins that user u_i has at location m_k with respect to the total number of check-ins at place m_k. Then $q_{1k}, ..., q_{Nk}$ is a discrete probability distribution that describes how likely a check-in at m_k was made by a certain user. Thus, we define E_k as the entropy of place m_k: $E_k = \sum_{u_i \in \Phi_k} -q_{ik}\ log\ q_{ik}$

Then we get features using E_k: *min_ent*, the minimum place entropy across all the shared venues, and *aa_ent*, the sum of the inverse of each place entropy value.

Above all, the paper gathered other feature based on place but can adopt on any pair. First, we define mli as the home where user ui has most check-ins. Then we get two features: geodist as the geographic distance between their home locations and *w_geodist* is the same distance divided by the product of the number of check-ins each user has done in their home location.

2) Graph Features: Except the features used in other papers, such as union and intersection of two nodes, we also add some well-known graph features, including jaccard coefficient and summation of inversed node degrees.

3) New Location Features: Based on the observation by "Friendship and Mobility: User Movement In Location-Based Social Networks", the more distance between a users check-in place and his home the more possibility it is correlated to his friends. Thus, we put weight on all place features above by multiply log of the distance from home on the number of check-ins a user to a place that is $w_{c_{ik}} = log(distance) * c_i k$.

D. Models

We try LibLinear, LibSVM, Random Forest (RF), and Gradient Boosting Model (GBM) in our link prediction task. However, the performance of LibLinear leaves much to be desired, and it is time-consuming to use LibSVM since the dual problem of this data is hard to solve. The detailed experiment results and figures are shown in section 5.

IV. Experiment Design

We first build some features recommended in papers, and judge its behavior with TA's dataset. However, we encounter a serious problem of enormous missing values. To slove this, we sample roughly the same number of positive and negative instances from source data. After that, we use knowledge of shared graph and checkin data to create features for sampled instances.

We compare different combination of features to find out the crucial factor that improve the prediction accuracy.

V. Experiment Results

A. Data given by TA

Because of the enormous missing of features, our models learn nearly nothing. The performance of all-zero prediction on *Gowalla* data is 0.62, whereas our model performance is about 0.64. Since the bad performance, we do not put any further experiments on these data.

B. Data Self-sampled for Gowalla

We put experiments on five dataset with five-fold cross-validation. They are:

- *O_Loc_F* + *Graph_F*: old checkin features+ graph features
- *Graph_F*: graph features
- *O_Loc_F*: old checkin features
- *ON_Loc_F*+*Graph_F*: old & new checkin features + graph features
- *ON_Loc_F*: old & new checkin features

1) RF: We use the default parameters in *scikit-learn* with 200 trees. Best performance 0.904296 appears when using features recommended in papers and our new created features. It is shown in [Table 1].

	O_Loc_F+Graph_F	Graph_F	O_Loc_F	ON_Loc_F+Graph_F	ON_Loc_F
TN	1.000000	0.992755	1.000000	1.000000	1.00000
TT	0.903322	0.870595	0.781290	0.904296	0.787047

TABLE I: RF Performance on Gowalla data

2) GBM: We grid search the parameters, and finally use parameter $n\_estimators = 800$ and $learning\_rate = 0.1$. Best performance 0.900638 appears when using features recommended in papers and our new created features. It is shown in [Table 2].

	O_Loc_F+Graph_F	Graph_F	O_Loc_F	ON_Loc_F+Graph_F	ON_Loc_F
TN	0.909475	0.889071	0.789329	0.910540	0.796655
TT	0.899927	0.881947	0.777462	0.900638	0.782721

TABLE II: GBM Performance on Gowalla data

3) Comparison: RF performs better than GBM generally. Figure is showned as [Fig. 3].

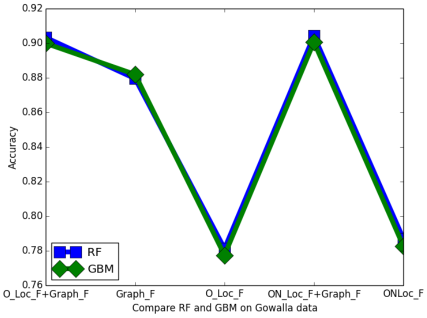

Fig. 3: Model Performance Comparison on Gowalla data

C. Data Self-sampled for Brightkite

1) RF: Same parameters as Gowalla. Best performance 0.846021 appears when using features recommended in papers and our new created features. It is shown in [Table 3].

	O_Loc_F+Graph_F	Graph_F	O_Loc_F	ON_Loc_F+Graph_F	ON_Loc_F
TN	1.000000	0.988453	0.999089	1.000000	0.999138
TT	0.843091	0.816264	0.698025	0.846021	0.712224

TABLE III: RF Performance on Brightkite data

2) GBM: Same parameters as Gowalla. Best performance 0.835197 appears when using features recommended in papers and our new created features. It is shown in [Table 4].

	O_Loc_F+Graph_F	Graph_F	O_Loc_F	ON_Loc_F+Graph_F	ON_Loc_F
TN	0.860425	0.837264	0.789329	0.863256	0.734836
TT	0.833875	0.818229	0.777462	0.835197	0.689112

TABLE IV: GBM Performance on Brightkite data

3) Comparison: The same conclusion as Gowalla. Figure is shown as [Fig. 4].

D. Stitched Data Self-sampled for Gowalla

1) RF: Similar to previous experiments. Best performance is 0.902819. It is shown in [Table 5].

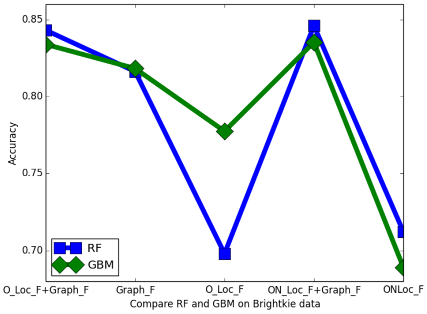

Fig. 4: Model Performance Comparison on Brightkite data

	O_Loc_F+Graph_F	Graph_F	O_Loc_F	ON_Loc_F+Graph_F	ON_Loc_F
TN	1.000000	0.992285	0.999995	1.000000	1.000000
TT	0.901394	0.877602	0.775497	0.902819	0.783377

TABLE V: RF Performance on Stitched Gowalla data

2) GBM: Similar to previous experiments. Best performance is 0.846021. It is shown in [Table 6].

3) Comparison: Similar to previous experiments. Figure is showned as [Fig. 5]

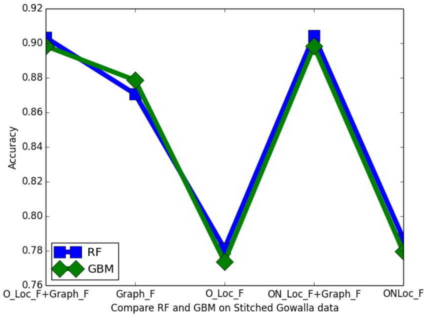

Fig. 5: Model Performance Comparison on Stitched Gowalla data

E. Stitched Data Self-sampled for Brightkite

1) RF: Similar to previous experiments. Best performance is 0.864703. It is shown in [Table 7].

2) GBM: Similar to previous experiments. Best performance is 0.860194. It is shown in [Table 8].

3) Comparison: Similar to previous experiments. Figure is shown as [Fig. 6].

VI. Discussion

A. New Location Features

From the experiment results, all models with new location features have a better performance. We can conclude that our distance-weighted features are useful for link prediction.

	O_Loc_F+Graph_F	Graph_F	O_Loc_F	ON_Loc_F+Graph_F	ON_Loc_F
TN	0.906684	0.885433	0.785831	0.907518	0.792430
TT	0.898379	0.878609	0.773823	0.898359	0.779512

TABLE VI: GBM Performance on Stitched Gowalla data

	O_Loc_F+Graph_F	Graph_F	O_Loc_F	ON_Loc_F+Graph_F	ON_Loc_F
TN	1.000000	0.994394	0.999703	1.000000	0.999709
TT	0.860581	0.835733	0.762521	0.864703	0.767030

TABLE VII: RF Performance on Stitched Brightkite data

	O_Loc_F+Graph_F	Graph_F	O_Loc_F	ON_Loc_F+Graph_F	ON_Loc_F
TN	0.892376	0.864218	0.804024	0.895903	0.808897
TT	0.859176	0.839079	0.762909	0.860194	0.763152

TABLE VIII: GBM Performance on Stitched Brightkite data

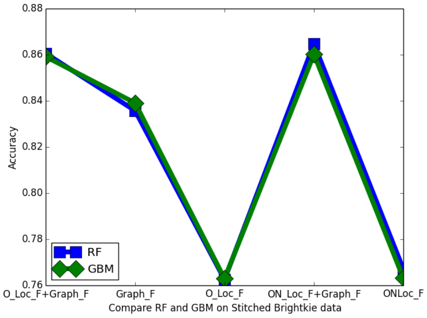

Fig. 6: Model Performance Comparison on Stitched Brightkite data

B. Stitched Location Data

The performance of stitched location data is not stable in our experiments. For *Gowalla*, the best performance drops about 0.2 percent. However, for *Brightkite*, the best performance enhances 3 percent, which is a quite significant improvement.

There are several possible factors that cause the unstability. First, it is hard to determine which r we should use. If r is too small, the stitching effect is not obvious, however, if it is too large, information for each user are blurred.

Furthermore, two places within distance r maybe the same location, maybe not. For example, users have checkins in the northern and southern part of Da-An Forest Park should use same location ID. Nevertheless, the northern part of park should not share the location ID with places across the street.

A better way to precisely solve this problem is to remap all locations into global map and categorize each location correctly, but we do not find any resources that we can use directly.

C. Time Factor

We do not put experiments on time factors of checkin data. Based on other parpers, using the time features will have slightly improvement on link prediction. This will be our toppest priority future work.

Appendix

D. Python code for location stitching

```
# Target: iteratively merge locations within distance r
# locInfo = locID: list (lat, lng)
# remapLoc = originLoc: mappedLoc
# calDis = function(locA, locB): dist
# findRoot = function(locID): parent locID

def mergeLocation():
  r = 0.001
  data = [[id, loc[0], loc[1]] for id, loc in locInfo.
       iteritems()]
  data = sorted(data, key = lambda inst: inst[2])
  flag = 0
  for i in range(len(data)):
    for j in range(flag, i):
      if (calDis(data[j][1:], data[i][1:]) < r):
        REMAP[findRoot(data[j][0])] = findRoot(data[i][0])
    while (flag < i and data[i][2] - data[flag][2] > r):
      flag += 1
```

References

[1] Ole J. Mengshoel, Raj Desai, Andrew Chen, and Brian Tran, *Will We Connect Again? Machine Learning for Link Prediction in Mobile Social Networks*, Eleventh Workshop on Mining and Learning with Graphs, .

[2] Salvatore Scellato, Anastasios Noulas, and Cecilia Mascolo, *Exploiting Place Features in Link Prediction on Location-based Social Networks*, KDD '11 Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining Pages 1046-1054 ACM New York, NY, USA 2011

[3] Amy X. Zhang, Anastasios Noulas, Salvatore Scellato, and Cecilia Mascolo, *Hoodsquare: Modeling and Recommending Neighborhoods in Location-based Social Networks*, SOCIALCOM, 2013, 2013 International Conference on Social Computing (SocialCom), 2013 International Conference on Social Computing (SocialCom) 2013, pp. 69-74, doi:10.1109/SocialCom 2013.17

[4] Eunjoon Cho, Seth A. Myers, Jure Leskovec, *Friendship and mobility: user movement in location-based social networks*, KDD '11 Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining Pages 1082-1090 ACM New York, NY, USA 2011