LOKASYON BAZLI SOSYAL AĞLARDA YER TAHMİNİ

Furkan KALE,Burak NUHBAŞA
Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi, 34220 Istanbul, Türkiye
{furkankale.1996, bhuhbasa}@gmail.com

***Özetçe* —Projemiz lokasyon bazlı sosyal ağlardaki kullanıcılar için konum tahmini yapmayı amaçlayan bir projedir. Bu projede lokasyon bazlı ağları kullanan kişilerin bir sonraki gidecekleri yeri başarılı bir şekilde ölçüp, kullanıcılara tavsiye ediyoruz.**

Lokasyon bazlı sosyal ağlarda kullanıcılara daha iyi bir konum tavsiyesinde bulunabilmek adına proje kapsamında çalışmalarımızı yürüttük.

Proje kapsamında kullanılabilecek 6 farklı yöntem var ve bu 6 temel yöntemi hibrit bir şekilde kullanıyoruz.Bu yöntemlere ilişkin detaylı bilgiler rapor içerisinde yer almaktadır.

Uygulamada 12.588 kullanıcı ile veritabanı kullanılmış ve %75 oranında başarı yakalanmıştır. Ayrıntılar raporun ilerleyen kısımlarında görülecektir.

Anahtar Kelimeler—Lokasyon bazlı sosyal ağlar, konum tahmini, veri madenciliği

***Abstract*—Our project aims to make position estimation for users in social network based location. In this project,we measure the next successor of those using social networks based location succesfully and recommend to the users.**

We have carried out our project work in order to find a better location advantage for users in location based social networks.

In the application,there are six different methods developed within the project and we used 6 basic methods in a hybrid way. Detailed information on these methods is included in the report.

12.588 users with database are used and 75 % percantage success are seen in the application. Details will appear later in the report.

Keywords—Location based social networks, location estimation, data mining

I. INTRODUCTION

In location-based social networks (LBSNs), users share information about their locations, the places they visit and their movement alongside with other social information. Visits are reported explicitly (by user check-ins in known venues and locations) or implicitly by allowing smartphone applications to report visited locations to the LBSN. This information is then shared with other users who are socially related (e.g., friends). The same information can be exploited by the LBSN operator to propose new points of interest to users. Recommending new locations is an important issue;it allows to efficiently advertise companies with a physical presence (theaters, bars, restaurants, etc.) and create revenue for the LBSN operator.

Recommender systems are widely used and they have been studied in research quite extensively. The most popular approach in recommender systems is that of collaborative filtering, where recommendations are created based on whether a user has purchased a product in the past and on whether she liked it or not. Using the past behavior of a user, new recommendations are created based on the similarity of users or the similarity of products (items). While these algorithms can be adjusted to the problem of recommending new locations to users, by taking into account previous user checkins,significant information like the distance of the proposed location to the user neighborhood or the social interaction between the user and those users that have visited this location are ignored.

Social networks have developed with the spread of the internetThere are many applications for different needs. Foursquare and Gowalla share their location and inform their friends. These applications are useful for the promotion of the place where the person is going and for the Friends of the person. Our work is to provide advice on where users are going and to do it as much as possible.Depending on the type of places visited, proximity,popularity, or places where friends went before the forecast will be made.

For the project, the database of the year 2010, which Gowalla opened to the public, will be used. Because this database is worldwide, it has many users, locations and locations. This can lead to overallocation and loss of time during the project's operation. Therefore, the database will first be reduced to a city and then the distinction will be made according to the types of spaces.

When estimating the location, the distance or popularity between the location will be taken into account.With GM-FCF algorithms, estimates will be made based on the user's history, the type of places and preferences. It is very important that the application has a high rate of accuracy and fast operation in the estimation while it is running. Inconsistent estimates can cause users to go to competing applications or leave such applications. The current database will be fully organized to meet the work to be done and also to work quickly.

II. COLLABORATIVE LOCATION RECOMMENDATION

Collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating).[1]

A. Random Walk and Restart

An ordinary random walk is a dynamic process run on a graph. The graph can be ad-dimensional integer lattice, and time may be discrete or continuous. In the simplest case where d=1, the random walk can move a step to the right or the left at a given time with a dened probability p. Thus, the process produces a non-deterministic array of state changes.

Presently, the problem is usually expressed as the sum of a sequence of independent and identically distributed (IID) random variables.Let Xi;i>=1 be a sequence of IID random variables, and let Sn=X1+X2+... +Xn.The integer-time stochastic process Sn;n1is called a random walk, or more precisely,a one-dimensional random walk based on Xi;i>=1.Though Sn is a simple random variable for any Sn;n>=1, the focus is not only on Sn, but also on the behavior of the entire random walk process. For example, for a given real number , the probability that a sequence will contain any term for which Sn, or in other words, the point of threshold crossing is a common question that can be answered through a random walk process.As a quick example, let's consider a simple random walk: if Xi are IID binary random variables with values 1 or -1, the probability Sn will take some value can be answered as follows. If m out of n trials are 1, then Sn=m-(n-m); Sn= 2*mn. If p is the probability of Xi= 1, then

Pr(Sn= 2*m-n) = n!/m!(n-m)!*pow(p,m)* (pow((1-p),(n-m)) The same concept can be extended to integer-valued random walks, where Xi are IID integer values instead of binary values

B. Geo-Measured Friend-based Collaborative Filtering

According to the spatio-social analysis of Foursquare data, we observe that nearby friends tend to share more commonly visited locations (see Figure 1(b)). Thus, instead of scanning friends' visited locations to calculate their similarity weights, an idea is to model the similarity weight between friends by their distance. Accordingly, we propose geo-measured friend- based collaborative filtering (GM-FCF) which uses linear regression method upon power-law distribution of distances between friends to learn a friend similarity model. The power-law distribution is formulated as y = x, where x denotes the variable of distance between friends and y here denotes the variable of common location ratio. Both y and x are often transformed into "log-log" scale, where a linear model can fit as below.

log10 y = w0 + w1 log10 x (2)

The original power-law distribution can be recovered via the following equation. Hence, we can simply apply a linear curve fitting method to realize regression as follows. More specifically, let y = log10 y and x= log10 x. We shall fit data as follows

y(x;W) = w0 + w1-x

where w0 and w1 are the linear coefficients, collectively denoted by W. In order to avoid over-fitting when we approach the weight coefficients by least square error method, we add a penalty term (i.e., regularization term) to discourage the coefficients from reaching large values as below.Where N presents the cardinality of input dataset, tn is the ground truth corresponding to x n, and y is the regularization term. As to be discussed further later in performance evaluation, we experimentally apply the above linear regression upon the Foursquare dataset and obtain which will be used in experiments. In GM-FCF, instead of scanning all locations to calculate the similarity weight between friends, we only access the latitude and longitude for similarity weight estimation. Thus, the computation cost for similarity weight calculation is estimated as 2 * |Ui|. The total computation cost is C(GM–FCF) = 2*|Ui|+m*|L|, where m * |L| accounts for the cost of missing rating predication. non-friend users do not have much value for reference in recommendation. As such, we only need to compute the similarity weight between friends, instead of all users, and the given user. We expect great savings in matrix computation because the matrix size is significantly reduced. There is a trade off in relying only on friends in collaborative recommendation. On the one hand, since non-friends are not considered, much noise is reduced and thus good for precision. On the other hand, there are indeed some cases where non-friends share common locations. Those persons also contribute to recommendation, especially when there are few similar users to the targeted person. Eliminating these nonfriends may hurt re- call as a result, i.e., some potentially preferred locations are not recommended due to the lack of nonfriends with similar location interests. We will study the impact on precision and recall later in the performance evaluation. In FCF, only friends are used as references in collaborative filtering. In other words, we only need to calculate the simi larity weight between friends, instead of every pair of users. Notice that we do not need to introduce social friendship explicitly to adjust similarity weight as in SCF. Since only friends are included to calculate the similarity weight wi;k, the social friendship has already been taken into consideration implicitly. For simplicity, we adopt cosine similarity measurement here. In the following, we redefine the rate predication function to consider only friends. Let U and L denote the user set and the location set in a location-based social networking system. Besides, let Ui denote the friend set for a given user ui. We assume that the system keeps track of the rating a user ui U put on a visited location lj L and denote it as ri;j . These recorded user ratings on locations are thus used to predict possible ratings of the user on unvisited locations. We denote this predication as bri;j and obtain this predicted rating of ui on lj as follows .

C. User-Based Collaborative Filtering

User-based collaborative filtering (UserCF) is based on the idea that similar users have similar preferences on locations.

To estimate the interest of u in location ', UserCF considers u1, u2, · · · , u|U| who have visited ' and see how similar they are to u. Intuitively if u is similar to most of these users, then it is highly possible that u will be interested in location too. This idea is easily applicable in our setting. For each user u we consider her visiting profile, and ignore

$$\text{br}i;j = \frac{\Sigma\ uk \in Ui\ rk;jwi;k}{\Sigma\ uk \in U\ i\ wi;k}$$

where U_i ($\subseteq U_i$) is the set of friends with top-m similarity weight. Notice that the number of friends ($|U_i|$) is usually much smaller than $|U|$, i.e. $|U_i| \ll |U|$.

In FCF, for a given user, only his friends are involved in the computation of similarity weight and contribute to the missing rating prediction. Thus, we estimate the computation cost for collaborative filtering as $C(FCF) = |U_i| \times |L| + m \times |L| < C(CF)$ because of $|U_i| \ll |U|$. Furthermore, we conclude that $C(FCF) < C(SCF)$.

any friendship information (i.e., EF edges in G are ignored). Based on the locations that u has visited and their visiting frequencies (v edges on the graph), we create the profile of user u as a vector pu of length |L|,

Pu =(wu,'1 ,wu,'2 , · · · ,wu,'|L|) Cosine similarity can be used to measure the similarity between the visiting profiles of two users. Therefore, the preference of user u on location ' can be estimated as After estimating u's preference on

$$\text{score}(u, \ell) = \frac{\sum_{u' \in U} \cos\left(p_u, p_{u'}\right) \cdot w_{u',\ell}}{\sum_{u' \in U} \cos\left(p_u, p_{u'}\right)}.$$

Figure 1 The Preference on User u on Location

all the unvisited locations, a recommendation list can be naturally generated by selecting the N locations with the highest scores

D. Friend-based Collaborative Filtering

Unlike previous methods which do not consider any social relationship, friend-based collaborative filtering (FriendCF) proposed in exploits the influence of friendships. FriendCF is based on the assumption that people listen to their friends and follow their friends' recommendations. FriendCF calculates the preference of a target user u on some location ' in a similar way to Eq. (4). The only difference is that FriendCF considers only u's direct friends Fu, i.e., the users that are directly associated to u with an friendship edge f in G instead of all users U. Thus the score in this case is given by the following equation:

$$\text{score}(u, \ell) = \frac{\sum_{u' \in F_u} \cos\left(p_u, p_{u'}\right) \cdot w_{u',\ell}}{\sum_{u' \in F_u} \cos\left(p_u, p_{u'}\right)}.$$

Figure 2 FCF Score

E. Location Nearest Neighbor

Motivated by the observation in Section 2.3 that users tend to visit places nearby their previous whereabouts, we create a simple reference recommender termed location nearest neighbor (LocNN) based only on the spatial distance of a location to the locations that have been previously visited by a user. Specifically, based on u's visiting history Lu = '1, '2, · · · , 'n _ L, score(u) of every 62 Lu is simply defined over the geographical distance between Lu:

$$\text{score}(u, \ell) = \frac{1}{\min_{\ell' \in L_u} \text{geodist}(\ell, \ell')}.$$

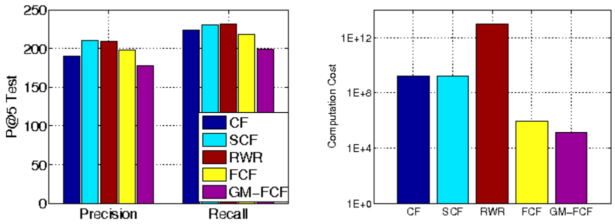
Figure 3 LNN Score

III. Dataset

Bright kite was an LBSN created in 2007 and closed after April 2012. The dataset contains 2,627,870 check-in records made by 58,228 users involving 772,933 locations over 942 days from 21 Mar 2008 to 18 Oct 2010. Among all the users there are in total 214,078 friendship links. Gowalla was an LBSN launched in 2007, purchased by Facebook in 2011, and closed in 2012. The dataset contains 6,264,203 check-in records made by 196,591 users involving 1,280,956 locations over a time period of 627 days from 04 Feb 2009 to 23 Oct 2010. The dataset also contains 950,327 friendship links among users.

IV. EXPERIMENTS AND RESULTS

Here we compare the overall performance of CF, SCF, FCF, GM-FCF and RWR4. Figure 2(a) plots both relative precision and recall for P@5. As shown, social friendship is beneficial for location recommendation in social networks, since all algorithms that take social relation into account outperform CF in terms of precision. However, in contrast to the significant improvement reported in [3], the social factor only brings minor improvement over the CF algorithm. As for the recall, SCF and RWR outperforms CF. FCF considers only friends in its recommendation, suffering missing knowledge of non-friend users and thus hurting its recall a little bit. However, the differences in recall amongst all compared algorithms are insignificant. The experimental result shows that the proposed FCF technique is very competitive in comparison with other collaborative recommendation techniques. On the other hand, the strength of FCF lies in its computational efficiency which is a mandate for on-line location recommendation. Figure 2(b) shows the computational cost (in log scale) among the compared techniques. As shown, FCF significantly outperforms all other techniques in several order of magnitude. In summary, FCF prevails because it only considers friends (a small number compared with the whole user set) when processing location recommendation for a given user, leading to a much lower computation overhead. On the other hand, the effectiveness of FCF remains competitive because nearby friends provide a high-quality pool of references for location recommendations.



(a) Effectiveness (b) Efficiency - log scale

Figure 4 Performance Comparision

V. FEASIBILITY TECNICAL

A. Hardware Feasibility

Hardware requirements of project are 4GB RAM and 2GB disk space.

B. Application Development Environment

Android Studio and DB Browser for SQLite

C. Operating System

Android

REFERENCES

[1] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo, "A random walk around the city: New venue recommendation in location-based social networks," in *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*. Ieee, 2012, pp. 144–153.