Online Friends Recommendation Based on Geographic Trajectories and Social Relations

Shi Feng[1,2], Dajun Huang[1], Kaisong Song[1], and Daling Wang[1,2]

[1] School of Information Science and Engineering, Northeastern University, China
[2] Key Laboratory of Medical Image Computing (Northeastern University),
Ministry of Education, Shenyang 110819, China
{fengshi,wangdaling}@ise.neu.edu.cn,
{huangdajun,songkaisongabc}@126.com

Abstract. With the rapid development of GPS-enabled mobile devices, people like to publish online data with geographic information. The traditional online friend recommendation methods usually focus on the shared interests, topics or social network links, but neglect the more and more important geographic information. In this paper, we focus on users' geographic trajectories that consisting of a series of positions in time order. We reduce the length of each trajectory by clustering the points and normalize every trajectory according to its positions and time in the trajectory. The similarity between trajectories is computed based on the distance of each corresponding point pair in the respective trajectory and the trajectories' trends. The potential online friends are recommended based on the trajectory similarity and social network structures. Extensive experiment results have validated the feasibility and effectiveness of our proposed approach.

Keywords: Friend Recommendation, Geographic Trajectory, Social Network.

1 Introduction

One major difference between virtual Web-based social network and real life social network is, new friends in real world tend to be geographically related. With the rapid development of GPS-enabled mobile devices, people like to publish online data, such as tweets, with geographic information, which have filled the gap between the virtual cyber world and real life world.

The traditional online friend recommendation methods usually focus on the shared interests, topics or social network links, but neglect the more and more important geographic information. There are potential similarities between users underlying in geographic information. For examples, people who have worked in a series of same cities or shared the same travel routes may be potential good friends.

How to apply users' geographic position information for friend recommendation is an important problem. Because a user does not always stay in one place, his (her) geographic position should change over time and form a trajectory. Recently, several papers have been published for computing the similarity between two users based on

H. Motoda et al. (Eds.): ADMA 2013, Part I, LNAI 8346, pp. 323–335, 2013.

their GPS data or positions [1,2,3]. However, most previous literatures only considered the different positions in the social network, but not the sequence of the positions with time labels. That means the trajectory (A, B, C) is regarded as the same as (C, B, A), when the two trajectories belong to two users respectively. As a matter of fact, the trajectory in time order may reflect certain personal habits that the position set neglects.

In this paper, we take the temporal order into account in user geographic position information and propose an approach of online friend recommendation method based on geographic trajectories and social relations. We regard a user's geographic trajectory just like a time sequence, consider the shape between time-ordered points as the trend of the trajectory or sequence, and compute the similarity of two time sequence trajectories by comparing both the distance between them and the trend of each sequence. The potential friends are recommended considering the trajectory similarity and the social network structures.

The rest of the paper is organized as follows. Section 2 describes geographic trajectory reduction method. Section 3 gives the algorithm of trajectory similarity based on distance and trend. Section 4 presents and discusses the experimental results. Section 5 introduces the related work, and the conclusion is given in Section 6.

2 Geographic Trajectory Reducing

2.1 Problem Definition

We first give out the following definitions for introducing the method of reducing geographic trajectory in detail.

Definition 1 (user position set). A user position set PS is a set consisting of geographic positions P, i.e. it is the position record of the user's traveling during a period of time. Here we denote it as PS=(P_1, P_2, …, P_m).

Definition 2 (user geographic trajectory). For a user position set, if every position in the set is in chronological order, we say the user positions form a geographic trajectory. For PS in Definition 1, we say PS is a user geographic trajectory T=(<P_1, t_1>, <P_2, t_2>, …, <P_i, t_i>, …, <P_m, t_m>), if the user stayed at P_i at time t_i, stayed at P_j at time t_j, and t_i is earlier than t_j ($i<j$ and i, j=1, 2, …, m).

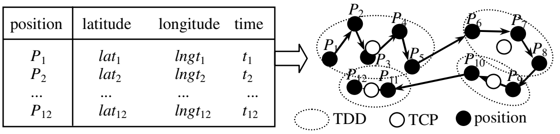
Definition 3 (trajectory dividing domain). Given the distance threshold and a certain period of time, the domain which is constituted by some positions within range is called trajectory dividing domain, and it is denoted as TDD.

Definition 4 (trajectory center point). The point which can represent the distribution of all points in the TDD it belongs to is called the trajectory center point, and it is denoted as TCP. In this paper, a TCP is the mean point of all points in the TDD it belongs to.

Definition 5 (**trajectory trend vector**). A user trajectory is a trajectory time sequence according to Definition 2. Here we define trajectory trend as the shape or style distributed on time of the trajectory or sequence. A user's trajectory trend vector's elements are composed of the slopes of two successive positions, denoted by $TTV=(slope_1, slope_2, \ldots, slope_{m-1})$, where $slope_i = \frac{P_{i+1}.lngt - P_i.lngt}{P_{i+1}.lat - P_i.lat}$ and $P_i.lat$, $P_i.lngt$ are the latitude and longitude of P_i, respectively.

2.2 Process of Geographic Trajectory Reducing

The purpose of reducing geographic trajectory is to compress a long geographic trajectory with many positions into a short geographic trajectory with relatively less positions by clustering the nearest positions in the geography. In this paper, each user's positions are represented by a time sequence. For each user's all positions, we cluster them by considering not only their latitude and longitude, but also the occurring time. The main idea of our method is shown in Figure 1.



position	latitude	longitude	time
P_1 P_2 ... P_{12}	lat_1 lat_2 ... lat_{12}	$lngt_1$ $lngt_2$ ... $lngt_{12}$	t_1 t_2 ... t_{12}

Fig. 1. Method of clustering geographic positions by latitude, longitude, and time

In Figure 1, if we do not consider the time, P_1 to P_5, P_{11} and P_{12} are clustered into one TDD because of their close latitude and longitude. However, in Figure 1, P_{11} and P_{12} are not clustered together with P_1, P_2, to P_5, because their time interval is too long. In next section, we measure the similarity between trajectories consisting of a series of positions according to their distance and trend. The trend is relevant to time, so we should consider the influence of time.

2.3 Algorithm of Geographic Trajectory Reducing

According to section 2.2, we consider not only the geographic positions but also the time of these positions during the clustering process. In detail, for a trajectory time sequence, we set a distance threshold α, then orderly compare these positions from the first position for clustering the sequence into some TDDs based on α. Finally, we use TCP to represent the corresponding TDD for every TDD. We give the algorithm of reducing geographic trajectory in Algorithm 1.

In Algorithm 1, a user's trajectory is represented with positions (P_1, P_2, …, P_n), where P_i (i=1, 2, …, n) is denoted by a triplet (t_i, lat_i, $lngt_i$), and the elements represent the time, latitude, and longitude respectively. They can be used to compute the distance between two positions in line 5). Resultant (TCP_1, TCP_2, …, TCP_k) is k TCPs of k TDDs, where k is not given beforehand as k-means, but is generated in the process of the reduction.

Algorithm 1: Geographic Trajectory Reducing

```
Input: A user's trajectory presented with ($P_1$, $P_2$, …, $P_m$), radius threshold $\alpha$;
Output: Refined the user's trajectory presented with ($TCP_1$, $TCP_2$, …, $TCP_k$);
Description:
  $i$=1;  \\ select the first position in trajectory
  $k$=0;
  Repeat
    $j$=$i$+1;  \\ select the next position for comparing
    while $j \le m$ and $|lat_i\text{-}lat_j| \le \alpha$ and $|lngt_i\text{-}lngt_j| \le \alpha$ do $j$=$j$+1;
    $k$=$k$+1;
    $TDD_k$={$P_i$, $P_{i+1}$, ..., $P_j$};  \\ build $k$th cluster $TDD_k$
    compute $TCP_k$ from $TDD_k$;  \\ compute the mean $TCP_k$ of $TDD_k$
    $i$=$j$+1;  \\ start a new clustering process
  Until $j$>$m$;
```

3 Trajectory Similarity Based on Distance and Trend

3.1 Normalization of Trajectory

After reducing every user trajectory, we discover that the number of each user's TCPs is not the same. In order to facilitate calculation, we have to normalize the number of all of users' TCPs as a fixed number. Assume that the number of each user's position before reducing is (p_1, p_2, …, p_n), and after reducing is (k_1, k_2, …, k_n), where n is the number of users. Our algorithm requires the number of two users' positions should be the same, so we can revise α according to the following iteration formulas until the number of each user's TCP equals $\bar{k} = \left\lfloor \frac{1}{n}\sum_{i=1}^{n} k_i \right\rfloor$.

For user i, his (her) number of position before reducing and TCP after reducing are p_i and k_i respectively, the jth iteration process is:

$$\hat{k}_i^j = Num(TDDs^j) \tag{1}$$

where $TDDs^j$ is the set of $TDDs$ at jth iteration, and $Num(TDDs^j)$ is the number of TDDs at jth iteration.

Let

$$\alpha^{j+1} = (\hat{k}_i^j / \bar{k}) \cdot \alpha^j \tag{2}$$

back to (1) for iteration, until $\hat{k}_i^j = \bar{k}$.

Note that it is difficult to get an ideal result straightly according to the above iteration formulas. In other words, we can't make the number of each user's TCPs reach $\bar{k}$. We try to consider the following two particular situations.

(1) When α^i=α^j occurs during the process of iteration, i.e. the threshold α of the ith iteration (namely α^i) is equal to the jth iteration (namely α^j). Here we assume i<j, then according to Formula (2) we have:

$$\alpha^j = (\hat{k}_i^j / \bar{k}) \times \alpha^{j\text{-}1} = (\hat{k}_i^{j-1} / \bar{k}) \times (\hat{k}_i^{j-2} / \bar{k}) \times \cdots \times (\hat{k}_i^i / \bar{k}) \times \alpha^i \qquad (3)$$

Since $\alpha^j=\alpha^j$, so we have:

$$(\hat{k}_i^{j-1} / \bar{k}) \times (\hat{k}_i^{j-2} / \bar{k}) \times \cdots \times (\hat{k}_i^i / \bar{k}) = 1 \qquad (4)$$

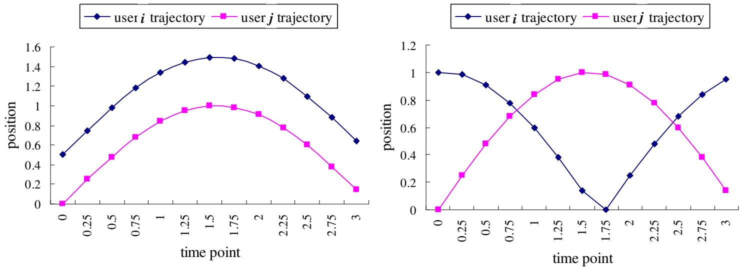
We can say that it is a period of *j-i* in the process of iteration. This will lead the iteration to an infinite loop and never get the results what we want. In order to solve this problem, we let a tag array to tag whether the iteration has been the state of infinite loop and a threshold β. After each iteration, we should determine whether Formula (4) is supported. If not, we use Formula (2) to continue our iteration. When Formula (4) is supported, we should use Formula (5):

$$\alpha^{j+1} = (k_i^j / \bar{k}) \cdot \alpha^j + \beta \qquad (5)$$

(2) Based on the situation that not all the final iteration of each user's TCPs can reach $\bar{k}$, we have to settle for the suboptimal results which is the closest to $\bar{k}$. In order to normalize the results and reduce error as far as possible, we put forward an additional iterative condition: the result of each user reaches to $\bar{k}$ from the *right side* of itself. Specifically, on the one hand, if the result of the user can precisely reach $\bar{k}$, the iteration will stop immediately. On the other hand, if the result of the user is higher than $\bar{k}$ and next iteration it will be smaller than $\bar{k}$, we can use it to tentatively represent the result. By the end of the iteration, we should have truncated those results which are higher than $\bar{k}$ and made them equal to $\bar{k}$.

3.2 Similarity Measure of Distance and Trend

The reason why we choose the distance and trend as the measures of two users' geographic trajectory similarity has two aspects. On the one hand, according to Definition 3, in a user's trajectory, the relation of context positions can infer the tendency of the trajectory sequence. On the other hand, for two users' trajectories, the distance of the corresponding position can intuitively reflect how near or far the two trajectories are. For two users *i* and *j*, there are two extreme situations shown in Figure 2.



(a) identical trend but far distance

(b) opposite trend but identical some positions

Fig. 2. Two extreme situations of distance and trend for two user geographic trajectories

In Figure 2, it's obvious that considering a single of any situation is not comprehensive. In Figure 2(a), the corresponding elements of the two trajectory trend vectors are identically same but they are far apart, while some corresponding elements of the two trajectory trend vectors are opposite but their distance can be close in Figure 2(b). So in this paper, we consider that combining the distance with trend for computing the similarity between trajectories.

Firstly, we discuss the similarity of distance. As we know, we can use distance to measure the similarity between two users. Some frequently-used distance includes absolute distance, Huasdorff distance, Minkowski distance, Mahalanobis distance, etc. Huasdorff distance is often used in collection which ignores the order of elements. Obviously, it's not suitable for this paper because we need to consider the time order. Minkowski distance (when p=1 it's Manhattan distance. When p=2 it's Euclidean distance. When $p=\infty$ it's Chebyshev distance) also has such drawback. Simplicity, for example, we put the factor of time aside temporarily and only take latitude and longitude into consideration. Assume that there are three people A(20,110), B(30,110) and C(20,120). No matter what's p equal to, the Minkowiski distance between A and B is equal to the Minkowski distance between A and C. In fact, 10° of latitude is not equal to 10° of longitude. The applicative condition of Mahalanobis distance seems too harsh, so it's also not suitable in this paper. In conclusion, the traditional methods of measuring the similarity of uses' geographic trajectory are not appropriate, thus we try to measure the similarity by the distance of latitude and longitude.

We can do some trigonometric transformation for the distance formula of longitude and latitude. Assume A (latitude: φ_1, longitude: θ_1), B (latitude: φ_2, longitude: θ_2), then we have $\cos(A,B)=A_1\times A_2+B_1\times B_2+C_1\times C_2$, where $A_1=\cos\varphi_1\times\sin\theta_1$, $B_1=\cos\varphi_1\times\cos\theta_1$, $C_1=\sin\varphi_1$, $A_2=\cos\varphi_2\times\sin\theta_2$, $\mathrm{B}_2=\cos\varphi_2\times\cos\theta_2$, $C_2=\sin\varphi_2$. Moreover, $dist(A,B)=R\times$ $\arccos(A,B)\times P_i/180$, where R is the radius of the earth. Since we only want to compare the similarity, the $\cos(A,B)$ can be the measurement. Assume that the geographic trajectory sequence of user i can be shown as $T_i=\{P_{i1}, P_{i2}, \ldots, P_{in}\}$, where $P_{ij}=<t_{ij}, lat_{ij}, lngt_{ij}>$ (j=1, 2, ..., n) and $n=\bar{k}$, lat denotes latitude and $lngt$ denotes longitude, then the distance similarity $simofdist$ between two trajectories of user i and j (their trajectories are denoted as T_i and T_j respectively) can be shown as:

$$simofdist\,(T_i,T_j)=\frac{1}{\bar{k}}\sum_{k=1}^{\bar{k}}\cos\Delta_k \tag{6}$$

where $\cos\Delta_k=\cos lat_{ik}\times\sin lngt_{ik}\times\cos lat_{jk}\times\sin lngt_{jk}$
$+\cos lat_{ik}\times\cos lngt_{ik}\times\cos lat_{jk}\times\cos lngt_{jk}+\sin lat_{ik}\times\sin lat_{jk}$

Next we discuss the selection of the trend. We can treat them as 2 dimension vectors, i.e. (latitude, longitude). The way to implement the above idea is mapping the TCPs to plane. Then we can choose the slope to depict the trend of the geographic trajectory sequence and get the slope eigenvector. At last, we compute the similarity of the slope vectors of users to represent the geographic trajectory similarity.

It's easy to get the eigenvector of user i, where $C_i=(l_{i,1}, l_{i,2}, \ldots, l_{i,n\text{-}1})$, $l_{i,j}=(lngt_{ij+1}\text{-}lngt_{ij})/(lat_{ij+1}\text{-}lat_{ij})$. For facilitating calculation, we handle with the initial eigenvector as follows:

$$C_i^{'}=(l_{i,1}^{'},l_{i,2}^{'},\cdots,l_{i,n-1}^{'}),\ \text{where}\quad l_{i,k}^{'}=\begin{cases}1 & if\ l'_{i,k}>0\\ 0 & if\ l'_{i,k}=0\\ -1 & if\ l'_{i,k}<0.\end{cases}$$

Then the trend similarity *simoftrend* of user *i* and *j* is:

$$simoftrend'(T_i,T_j)=\frac{\sum_{k=1}^{\bar{k}-1} l'_{i,k}\times l'_{j,k}}{\sqrt{\sum_{k=1}^{\bar{k}-1}(l'_{i,k})^2}+\sqrt{\sum_{k=1}^{\bar{k}-1}(l'_{j,k})^2}} \tag{7}$$

The range of values of the trend similarity that gets from Formula (7) is [-1, 1], which is not consistent with the traditional distance formulas, because the range of values of the traditional is [0, 1]. In fact, the result is just a right feedback to our method of handling with the location data. As we said before, we regard users' position data as vectors, so the results may not only be positive, but also be negative. It seems entirely reasonable. Specifically, the positive indicates the tendency of the geographic trajectory of the two users' are in the same direction, while the negative indicates the opposite direction. In order to combine with the similarity of distance better, we can also standardize the similarity of the characteristic.

$$simoftrend(T_i,T_j)=\frac{1+simoftrend'(T_i,T_j)}{2} \tag{8}$$

We have obtained the similarity of the distance and trend respectively. Next we need to introduce a parameter λ as weight to represent how the two kinds of similarity prorate in the geographic trajectory similarity. We will give the value of λ by experiment. In a word, the geographic trajectory similarity of the user *i* and *j* is given as follows:

$$sim(T_i,T_j)=(1-\lambda)simofdist(T_i,T_j)+\lambda simoftrend(T_i,T_j) \tag{9}$$

According to Formula (6), (7), (8), and (9), we give the algorithm of computing geographic trajectory similarity based on distance and trend as Algorithm 2.

Algorithm 2: Measuring Trajectory Similarity

Input：two user trajectories reduced: T_i, T_j;
Output: similarity between T_i and T_j: *sim*(T_i,T_j);
Description:

```
1) For every user trajectory $T_i$ and $T_j$
2) {Repeat;  \\ normalize trajectories reduced, i.e. $T_i$ and $T_j$
3)     adjust the threshold $\alpha$ according to Formula (2);
4)     If there is any cycle Then take Formula (5);
5)     back to Formula (1);
6)   Until $\hat{k}_i$ ($\hat{k}_j$) is nearest to $\bar{k}$ and $\hat{k}_i$ ($\hat{k}_j$)$\geq\bar{k}$ ;
7)   compute distance similarity simofdist($T_i$,$T_j$) with Formula (6);
8)   compute trend similarity simoftrend($T_i$,$T_j$) with Formula (7) and (8);
9)   compute trajectory similarity sim($T_i$,$T_j$) with Formula (9);
10) }
```

For online friend recommendation, firstly we consider the social network structures to find the potential friends. Then the trajectory similarity is calculated to locate the most relevant friends of a candidate user.

4 Experiment

In this paper, the experimental data comes from a certain position service community users and Sina microblog users, where Sina microblog's data is the geographic positions of "second degree" friends of 100. Here "second degree" friends mean friends of friends. We choose 100 "second degree" friends for each user. Here we use the data of the position service community to verify that our algorithm is feasible and use the data of Sina microblog to show its practical significance. Choosing the geographic positions of "second degree" friends is to weaken the influence of the social network structure. If directly choose the user's friends, it will be inevitably blend the subjective factors into the results and reduce the credibility of the experimental results. The core of the traditional algorithm of friend recommendation is whether he/she is friend's friend. It shows that the user's "second degree" friends are the potential friends. Thus choosing the geographic positions of "second degree" friends in this paper not only takes the potential factor, but also weakens the impact of the social network structure as far as possible.

4.1 Results of User Trajectory Reduction and Normalization

In setting threshold α, on the one hand, if we set α a too high value, the number of some especial users' (their positions are almost exactly the same no matter at any time, i.e. they belong to the "Otaku") positions will possibly reduce to 1. Obviously, this is infeasible. On the other hand, if we set α a too small value, then the number will change slightly and it can't achieve the result of reducing the data. In the experiment of reduction, we set threshold α=1. After reducing, the geographic trajectory information of a user' 100 "second degree" friends is shown as Figure 3.

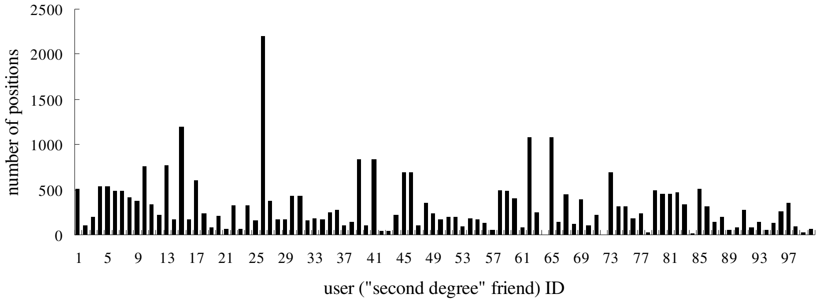


Fig. 3. Position number in 100 user trajectories after reducing the trajectories

We find that the number of each “second degree” friend’s geographic positions is very different. In fact, this can be explained by the fact that different people have different routine patterns. Some people are active while others are male/female “otaku”. For example, the number of geographic locations of user 26 still reaches up to 2199, while the user 72 has only 1. This can show that user 26 is more active than user 72 during this period of time. We mine this kind of information to determine whether the user is active or not. Obviously, here we can say that the user 26 is an active user and the user 72 is a typical male/female “otaku”.

Next we normalize the trajectories. Here we set β=0.001. The position number (i.e. TCP number in Definition 4) of a user’ 100 “second degree” friends after normalizing is shown as Figure 4.

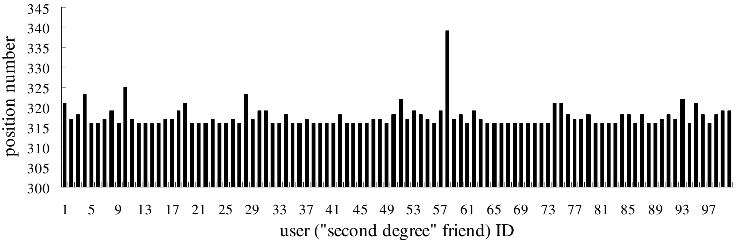


Fig. 4. Position number in 100 user trajectories after normalizing the trajectories

From Figure 4 we find that most of the number of the positions of “second degree” friends is equal to 316. For those which are not equal to 316, we only need truncate the portion which is higher than 316. Of course, we admit that we could face with the errors.

4.2 Similarity Computation Methods Comparison

In order to verify the feasibility of the algorithm proposed in this paper, we compare the proposed algorithm with the algorithm of use cosine formula to compute the similarity after DBSCAN clustering. Experimental data comes from the 100 users’ locations in a certain location service community, and the number of each user's geographic location is greater than 3410.

To compute the accuracy of the algorithm we adopt 10-cross validation method which divides equally 100 users’ location data into 10 portions (1~10, 11~20, ..., 91~100) and treats 9 of 10 as a training set, the rest one as a test set, then uses the mean of all of the results to estimate the accuracy of the algorithm. The main purpose of the training model is to determine the value of the parameter, here, we will use the Least Squares (LS) to determine the value of λ.

$$\begin{aligned}&\arg\min_{\lambda}\sum_{i=1}^{90}\sum_{j=i+1}^{90}[sim(T_i,T_j)-simofDBSCAN(T_i,T_j)]^2\\&=\arg\min_{\lambda}\sum_{i=1}^{90}\sum_{j=i+1}^{90}[(1-\lambda)simofdist(T_i,T_j)+\lambda simoftrend(T_i,T_j)-simofDBSCAN(T_i,T_j)^2\end{aligned}\qquad(10)$$

where *SimofDBSCAN*(T_i,T_j) is the similarity between trajectory T_i and T_j for user *i and j* by the algorithm of using DBSCAN clustering. Then, we calculate the accuracy of our algorithm, the formula is as follows:

$$Accuracy = 1 - \frac{1}{\sum_{i=1}^{10}\sum_{j=i+1}^{10} j}\sum_{i=1}^{10}\sum_{j=i+1}^{10} | sim(T_i, T_j) - simofDBSCAN(T_i, T_j) | \tag{11}$$

The value of *simofdist*(T_i,T_j), *simoftrend*(T_i,T_j) and *simofDBSCAN*(T_i,T_j) of partial users in a certain process of training the model is shown as Figure 5.

From Figure 5, we can find that in addition to the individual points, the values of the *simofDBSCAN*(T_i,T_j) are basically between the values of the *simofdist*(T_i,T_j) and the values of the *simoftrend*(T_i,T_j). This shows that our algorithm is feasible. A more in-depth mining, we find that the values of the *simofDBSCAN*(T_i,T_j) are closer to the values of the *simofdist*(T_i,T_j). Table 1 gives the result of each cross validation based on Formula (10) and the accuracy based on Formula (11). The average of λ is 0.31 and the average of the accuracy is 85.927%.

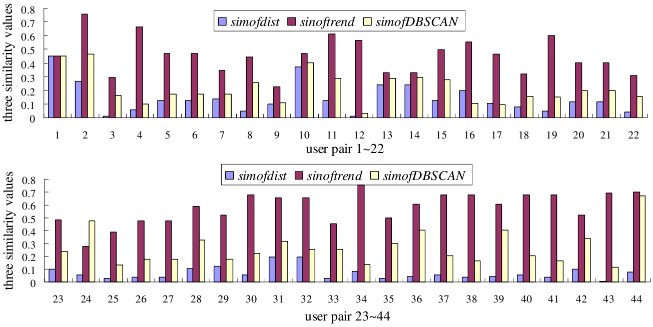


Fig. 5. Comparison of simofdist(T_i,T_j), simoftrend(T_i,T_j) and simofDBSCAN(T_i,T_j)

Table 1. λ and accuracy in every cross validation

Training ID	1	2	3	4	5	6	7	8	9	10
λ	0.29	0.33	0.21	0.39	0.27	0.37	0.41	0.28	0.35	0.20
Accuracy	0.8907	0.8473	0.8701	0.8284	0.8674	0.8143	0.8521	0.8365	0.8745	0.9114

4.3 Online Voting Activity

We launch an online voting activity and say "Now there will be an activity in your city, please choose the top-10 *second degree* friends whom you most want to invite to go with you". The voter determine whom will be invited by observing the each of "second degree" friend's personal information and the geographic locations recently emerged in Sina Weibo (It does not need to observe the latitude and longitude directly, but need to visually see the information of nation, province/district, street, etc.). Here we give the formula of computing the precision.

$$precision = \frac{NumofTop[1,10]}{10} + 0.3\frac{NumofTop[11,40]}{10} + 0.2\frac{NumofTop[41,70]}{10} + 0.1\frac{NumofTop[71,100]}{10} \qquad (12)$$

The reason why we definite the precision with Formula (12) is that the expectation of adoption rate in the case that voting 10 from 100 users is 0.1 if and only if when the voted top-10 "second degree" friends are just the numbers of the range from 71 to 100. We know that the similarity between the voter and his/her "second degree" friends will be determined when maximizing the precision. In our experiment, what we want to get is the value of the parameter λ. We can obtain the information that which kind of similarity plays a more important role in our voting activity.

Our experiment chooses 100 voters. We can compute each user's precision and the average precision of all 100 voters whenever the parameter λ takes a specific value. Figure 6 plots the average precision of all 100 voters as the parameter λ changes.

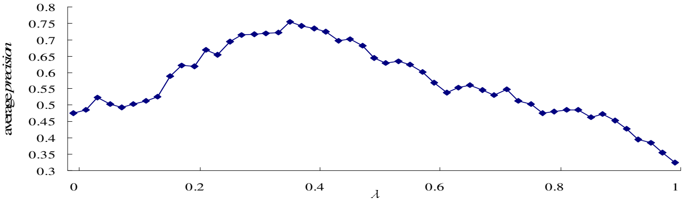


Fig. 6. Average precision of all 100 voters as the parameter λ changes

We find that the average precision up to the maximum value when λ is about 0.38. That is, when the average precision up to the maximum value, the distance similarity is more important than the trend similarity. In fact, the result is acceptable, because our experiment of voting does not impose strict restrictions on the order of the location and the voters consider more about the factor of distance.

5 Related Work

Researching on the geographic position information is very popular in recent years. Knowledge of users' positions can help improve large scale systems, such as cloud computing [12], content-based delivery networks [4], and location-based recommendations [5, 6, 7]. Additionally, the research work based on GPS data of Microsoft Research Asia has made achievements which can fix academia's eyes. The proposed methods include several stages: 1) Use GPS data to do some simple mining, such as the traveling ways [8], mining users' similarity based on trajectory [9, 10], understanding users' behaviors, and mining interesting location [11]. 2) Provide LBS service based on cloud by combining with external information [12]. 3) Achieve the recommendation of locations and activities by combining collaborative filtering algorithm [6, 13]. 4) GPS data is close to real life and also service real life[14].

Our work is quite different from above work. Firstly, we reduce user trajectories according to the order of time and unify them. Secondly, when computing the similarity between trajectories, we consider not only the distance between the trajectories but also the trend of every trajectory.

6 Conclusion

In this paper, we propose an online friend recommendation approach based on geographic trajectory similarity and social relations. We regard the trajectory consisting of a series of positions as a time sequence. We reduce the trajectory based on position and time order, and use the precise distance of latitude and longitude as the measure of the distance similarity and the slope as the measure of trend similarity. Finally, the similarity between trajectories is the result of weighted combination on distance similarity and trend similarity.

The distance similarity and trend similarity will play a different weighted role in different applications respectively. We propose a dynamic similarity formula. It's feasible to identify the weights of distance similarity and trend similarity through practical application and experiment. Certainly, for verifying the validity of the proposed algorithm, we have weakened the social network structure. We believe if combining the geographic position information with the social network structure, more ideal result will be got in applications. This will be our future work.

Acknowledgments. This research is supported by the State Key Development Program for Basic Research of China (Grant No. 2011CB302200-G), State Key Program of National Natural Science of China (Grant No. 61033007), National Natural Science Foundation of China (Grant No. 61100026, 61370074), and the Fundamental Research Funds for the Central Universities (N120404007, N100704001).

References

1. Yu, X., An, A., Tang, L., Li, Z., Han, J.: Geo-Friends Recommendation in GPS-based Cyber-physical Social Network. In: ASONAM 2011, pp. 361–368 (2011)
2. Ye, Y., Zheng, Y., Chen, Y., Feng, J., Xie, X.: Mining Individual Life Pattern Based on Location History. In: Mobile Data Management, pp. 1–10 (2009)
3. Ying, J., Lu, E., Lee, W., Weng, T., Tseng, V.: Mining user similarity from semantic trajectories. In: GIS-LBSN, pp. 19–26 (2010)
4. Leighton, T.: Improving Performance on the Internet. ACM Queue (QUEUE) 6(6), 20–29 (2008)
5. Hao, Q., Cai, R., Wang, C., Xiao, R., Yang, J., Pang, Y., Zhang, L.: Equip tourists with knowledge mined from travelogues. In: WWW 2010, pp. 401–410 (2010)
6. Zheng, V., Zheng, Y., Xie, X., Yang, Q.: Collaborative location and activity recommendations with GPS history data. In: WWW 2010, pp. 1029–1038 (2010)
7. Zheng, Y., Zhang, L., Xie, X., Ma, W.: Mining interesting locations and travel sequences from GPS trajectories. In: WWW 2009, pp. 791–800 (2009)

8. Zheng, Y., Liu, L., Wang, L., Xie, X.: Learning transportation mode from raw gps data for geographic applications on the web. In: WWW 2008, pp. 247–256 (2008)
9. Zheng, Y., Chen, Y., Li, Q., Xie, X., Ma, W.: Understanding transportation modes based on GPS data for web applications. TWEB 4(1) (2010)
10. Li, Q., Zheng, Y., Xie, X., Chen, Y., Liu, W., Ma, W.: Mining user similarity based on location history. In: GIS 2008, vol. 34 (2008)
11. Xie, X., Zhang, Y.: Understanding User Behavior Geospatially (November 30, 2008), `http://research.microsoft.com/apps/pubs/?id=74370`
12. Yuan, J., Zheng, Y., Xie, X., Sun, G.: Driving with knowledge from the physical world. In: KDD 2011, pp. 316–324 (2011)
13. Herlocker, J., Konstan, J., Terveen, L., Riedl, J.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. (TOIS) 22(1), 5–53 (2004)
14. Yuan, J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., Huang, Y.: T-drive: driving directions based on taxi trajectories. In: GIS 2010, pp. 99–108 (2010)