A USER BEHAVIOR PREDICTION AWARE COLLABORATIVE RECOMMENDATION ALGORITHM FOR LOCATION BASED SERVICES

Mingjin.Xin, Shunxiang.Li, Liyuan.Zhou, Weimin.Li

School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China

xinmj@shu.edu.cn

Abstract

Nowadays, location based services (LBS) has become one of the most popular applications with the rapid development of mobile Internet environment. More and more research is focused on discovering the required services among massive information according to the personalized behavior. In this paper, a collaborative filtering (CF) recommendation algorithm is presented based on the Location-aware Hidden Markov Model (LHMM). This approach includes three main stages. First, it clusters users by making a pattern similarity calculation of their historical check-in data. Then, it establishes the location-aware transfer matrix so as to get the next most likely service. Furthermore, it integrates the generated LHMM, user's score and interest migration into the traditional CF algorithm to generate a final recommendation list. The LHMM-based CF algorithm mixes the geographic factors and personalized behavior and experimental results show that it has more accuracy than other state-of-the-arts algorithm.

Key words: behavior prediction; LBS; collaborative recommendation; LHMM

1. INTRODUCTION

With the rapid development of mobile Internet and spatial information processing technology, user's behavioral sequences make up a tremendous location database and attract considerable research interests.

Under the mobile Internet environment, collaborative filtering (CF) is an effective recommendation algorithm. But when it applies to the behavioral prediction, it has limitation. First, user may visit a shop repeatedly, while traditional CF algorithm, like Netflix movies rating, just recommends item once. Second, users' next action is greatly depends on their former choice, which traditional CF doesn't take continuity into consideration. That is to say, in order to deal with services transition, traditional CF algorithm is not as good as Hidden Markov Model (HMM), which is frequently used to predict the probability of services-to-service transfer.

Through mapping the geographic information and service categories, the location-aware HMM (LHMM) will be constructed in this paper. This model gives the occurring probability of each service, along with the most likely occurred area. The new proposed approach, which combines LHMM and CF, is called CF-Behavior prediction algorithm. It can not only reduce the dimensions of similarity calculation, but also make a clear classification in attributes of different services within a similar behavior user groups.

The new LHMM model both consider the location and personalization factor. Initially, based on the idea of ClustalW algorithm, the improved method is suitable for handling sequences match in multiple groups. After extracting data from users who own the similar routine activities, a LHMM-based service to service transition probability matrix will be generated by integrating the location and service information into HMM training algorithm: weighted Baum-Welch. Finally, recommendation list for an active user is created by CF-Behavior prediction

algorithm. It takes user's score, check in frequency and interest migration into account.

2. RELATED WORK AND PROBLEM STATEMENT

In this section, some existing research works on HMM prediction and on check in data mining will be reviewed as below.

As for HMM strategies, (Blasiak, 2011) applied HMM to the classification, which completed the sequence classification by combining Baum-Welch, Gibbs sampling and change function together. (Antwarg, 2012) used multiple HMM to distinguish user from different devices, ages and gender. Thus, it could provide a more accurate prediction of user's next behavior. With the growing popularity of mobile terminals, HMM also began to be applied to the field of path analysis. (Hamada, 2013) used a modified BP-AR-HMM algorithm to predict user's driving behavior under multi-time series. (Wesley, 2012) converted earth into a plurality of triangular surface, and completed the prediction of user's next location through putting labeled triangle into HMM learning model.

As for LBS recommendation, (Lemire, 2005) proposed a very efficient algorithm-slope one, which made query more efficient, matrix more scalable than before. (Yuan, 2013) reduced the dimension of check in data by category, and then used the point of interest (POI) to calculate of similarity between users. (Ye, 2013) used check in data from Gowalla, and put into MCMC model to predict user's next behavior. The user's next possible activities can be predicted in accordance with the transfer matrix.

There are many studies on HMM improvement and optimization, like the reduction on high complexity and non-personalization (Zukerman, 1999). However, under the mobile recommendation environment, few researches

consider a good solution on processing spatial data, as well as gives a personalized prediction. Traditional CF algorithm used to process single-service, the new improved framework enables multi-service recommendation.

3. BEHAVIORAL SEQUENCE PREDICTION

In the following sections, detailed introduction to the framework and its related algorithm are presented respectively.

3.1 Behavioral prediction and recommendation framework

As shown in Fig.1, the whole recommendation framework is divided into three key modules: similar behavior cluster, series forecast and CF-behavior prediction.

- (1) **Behavior cluster:** This module is design to group users who enjoy similar life rhythm together. It will generate the top-k users who have similar life pattern.
- (2) **Series forecast:** This module trains similar user's check in records and obtain three main parameters of LHMM. The parameters are defined as initial probability matrix, transition probability matrix and emission probability matrix respectively.
- (3) **CF-behavior prediction:** Generally, users with similar behavior don't necessarily have similar interests. The purpose of this module is to give a more detailed recommendation, by combining the users' rate, visit frequency and interest migration.

The three modules build a coherent process, where the output of previous module will be the next module's input. Once check-in data have been put into behavior cluster

Fig. 1. The BP model attempts to predict behavior and recommend services

module, it will generate the top-k users who have similar life pattern. Then, using this users group can process the behavior transfer matrix in series forecast module. After calculating the max probability action user can take in the next step, a final recommendation list will be shown, which is generated according to users' interest.

3.2 Behavior sequence model

The core concept of our theory is to calculate the transfer probability between two time states for different user groups. For example, users often check in at a restaurant between 11:00 and 13:00, while they may take another activity between 13:00 and 17:00.

In order to reduce the sparseness, the check in data is then to be divided into six different stages, namely {1-5, 6-10, 11-13, 14-16, 17-19, 20-24}.

Definition 1 (Check-in): Check-in (*CK*) indicates that a user *U* check in at shop *S* at a certain time *T*. User rates shop *S* with *R*, $R \in \{0, 10, 20, 30, 40, 50\}$.

$CK = (\text{userId}, \text{userName}, \text{user City}, \text{time}, \text{shopId}, \text{star}, \text{comment})$

Definition 2 (Shop): Shop *S* represents the place where user participates in an activity with check-in.

$\text{Shop} = (\text{shopId}, \text{shopName}, \text{address}, \text{city}, \text{district}, \text{area}, \text{category}, \text{subcat}, \text{lat}, \text{lon})$

A broad description of shop's business scope is described as category, such as food, recreation, shopping, etc. While a narrow description is called subcategory, for

example, food is divided into Chinese cuisine, western, desserts or other subcategories.

3.3 Behavioral sequence similarity calculation

Based on the two definitions mentioned above, a check in record provides useful information at different perspective and level. A tree shaped undirected graph is established to describe the relationship between location and category, which is shown in Figure 2. D represents district of a city. S represents the service category. A represents a popular area in a district. For example, "-", 170201, 170101, -, -, -, " means a user takes part in a recreation activity in Baoshan district at 6:00-10:00 in the morning, and dining at 11:00-13:00 in the same district.

Fig. 2. A graph shows the relationship among district, service and associated area

Definition 3 (Score Function) : x and y are two check in records. $\sigma(x, y)$ represents the output through function σ . Each record contains three parts of information, which represent district, service category and popular area. Consider that changing district is more difficult than changing service or area, so the bonus points are designed as follow:

$$\sigma = \begin{cases} \sigma + 4 & x.\text{district} = y.\text{district} \\ \sigma + 2 & x.\text{service} = y.\text{service} \\ \sigma + 1 & x.\text{area} = y.\text{area} \\ \sigma - 1 & x \neq y \end{cases} \quad (1)$$

Definition 4 (Sequence Similarity): Given two sequences $S=S_1 \rightarrow \dots \rightarrow S_n$, $T=T_1 \rightarrow \dots \rightarrow T_n$. S_i represents the i th check in record in sequence S . $|S|$ denotes the length of the sequence S . The similarity between two behavioral sequences is calculated by Equation 2 as below:

$$\text{Score} = \sum_{i=1}^m \sigma(S_i, T_i) \quad \text{where } m = \min(|S|, |T|) \quad (2)$$

Definition 5 (Behavior Similarity): User $U1$ has multiple sequences. So does user $U2$. $A(U1, U2)$ shows the highest similarity after a one-to-one comparison between two users.

ClustalW(Thompson, 2002) is used for carrying out automatic multiple alignment of nucleotide or amino acid sequences. It constructs a distance matrix to show the relationship between two reaction sequences. Based on ClustalW sequences match theory, the sequences similarity one-on-one between two different groups can be calculated. The pseudo-code of this algorithm is listed to solve multi-groups similarity match as follow:

Algorithm 1. ClustalW based multi group multi sequence similarity, CB-MGMS

Input: $S=S_1S_2\dots S_m$, $T=T_1T_2\dots T_n$
Output: $\text{Int } A(S, T)$ // similarity of S and T
Algorithm:
 new $\text{Int}[m][n]$ result;
 for $i=1$ to n do for $j=1$ to m do // data prepare
 $A[i] = \text{split}(S[i])$; $B[j] = \text{split}(T[j])$ // parsing a sequence to several check in record
 For $k=1$ to 6 do // Analyzing similarity at each time period
 $\text{Result}[i][j] = \sigma(A[k], B[k]) / (\min(\text{len}(S), \text{len}(T)) * 7)$ // obtain check in record similarity
 // and do normalization operation
 For $k=1$ to $\min(n, m)$ do // terminate loop after a maximum similarity value has been found
 For $i=1$ to n do for $j=1$ to m do
 $\{\text{maxValue}, \text{maxPosI}, \text{maxPosJ}\} = \text{findMaxInMatrix}()$ // record the max value position
 $\text{VisitedI}[\text{maxPosI}] = \text{true}$; $\text{VisitedJ}[\text{maxPosJ}] = \text{true}$;
 // mark the row and column where max value is as visited
 $\text{SimValue} += \text{maxValue}$ // Cumulative similarity value
 $\text{simValue} = \text{simValue} / \min(a.\text{length}, b.\text{length})$

First of all, an $m*n$ matrix is constructed, where m represents the total sequences count of user S while n is the total number of sequences of user T . Furthermore, the similarity for every records of user S with the records of user T is calculated by applying the definition shown in Equation 1.

The time complexity of two users' similarity calculation is $O(n^2)$. n is based on the minimum sequences count of a user. Generally speaking, once system collects a new sequence of a user, the algorithm needs to be run to recalculate the similarity to meet the new requirements for the user.

4. BEHAVIORAL PREDICTION

4.1 LHMM (Location-aware Hidden Markov Model) generation

Through mining the location-based behavior sequences, users who own similar life rhythm can be easily clustered under the mobile service environment. Based on the sequential matching degree, a top-K location-aware HMM model (TopK-LHMM) is proposed to makes the location-based service prediction more intelligent.

A set of hidden states $S=\{S_1, S_2, \dots, S_M\}$ represents user's current location, along with a set of observations $O=\{O_1, O_2, \dots, O_N\}$ which represents the activities participated by the users. Figure 3 shows an illustration of relationship between hidden states and observations. There are three important parameters defined as follows:

- (1) The initial probability matrix indicates the probability of each hidden state $S_i \in S$.
- (2) Transition probability matrix indicates the probability from hidden state S_i to hidden state S_j .
- (3) Emission probability matrix indicates the probability of observed $O_i \in O$ under a given state S_i .

Fig. 3. The relationships between hidden states and observations of LHMM

Where O_i represents a check-in record. S_i denotes the location where this activity is taken place. If a user checks in multiple times within a day, a sequence will be built as $CK=O_1O_2..O_T$.

In order to search a set of right LHMM parameters to maximize the value of $P(O|\lambda)$, expectation-maximization algorithm (EM) is used here. (Welch, 2003) also known as forward-backward algorithm, is the most widely used method for solving HMM learning. The basic idea is to use a random initialization λ assuming that this λ is the optimal solution.

What's more, in view of occurrence frequency of the sequence, we improve the Equation 8-10 listed in paper (Ingrid, 1999) with weight factor, so that periodic sequences can be better handled in new model. The improved formula is shown below:

$$L(i) = \sum_{0 \leq k \leq L(i)} \gamma_{i,0}(1) * C(0_k) = \frac{L(i)}{\sum_{0 \leq k \leq L(i)} 1} \quad (3)$$

$$M(i, j) = \frac{\sum_{k=0}^{L-1} \sum_{l=0}^{L-1} \varepsilon_{i,j0}(t) * C(O_k) \frac{M(i, j)}{\sum_{k=0}^{L-1} \sum_{l=0}^{L-1} M(i, j)}}{\sum_{k=0}^{L-1} \sum_{l=0}^{L-1} M(i, j)} \quad (4)$$

$$\frac{N(\frac{1}{p}, \frac{1}{q}, \frac{1}{r})}{N(i, E(\frac{1}{p}))} \sum_{0 \leq k \leq \frac{1}{p}} \sum_{t=1}^L \delta_{0, \frac{1}{p} + \frac{1}{q} + \frac{1}{r}} \gamma_t(t) * C(0_k) \frac{B_1(E_p)}{B_1(E_p)} \quad (5)$$

Where $C(\text{ㄱㄷㅅ})$ represents the weight of observed sequence ㄱㄷㅅ , 행_i represents the probability of generating sequence Z under the state i , 행_{ij} shows the probability of generating sequence Z during the transition from state i to state j . The original equation multiplied by the weight $C(\text{ㄱㄷㅅ})$, L , M , N formula is obtained which help specify the certain occurred frequency of sequences.

Algorithm 2 is the pseudo-code of Top K-LHMM algorithm. Here we proposed several key steps, such as choosing k users who are the closest to target user U and putting their historical check in records as observation sequences into LHMM to train. Iteration will be terminated once reaching the error threshold, or running out the iteration times.

Algorithm 2. *TopK-LHMM*

Input: User: U , Behavioral Similarity Matrix : $SimBehav$

Output: HMM parameters

Algorithm:

```

IDs = getTopSimilarUserId(U, SimBehav, K) // top k similar
users of U

```

```
for ID in IDs do // assign check in info to array ObservedSeq
```

and HiddenSeq

```
ObservedSeq.append(checkRecordToObserveSeq(ID))
```

```
sequenceWeight.update(seqInd) // Record observations
sequence
```

```
weightHiddenSeq.append(checkRecordToHiddenSeq(ID))
```

```
Lamda = randomGenerate(pi,A,B) // randomly generated
initial parameters lamda
```

```
For I in ObservedSeq do // scoreCal(), purpose is to iteration
until satisfies the threshold
```

```
Score += prob(I,lamda) // calculate the probability of each
servation under this lamda model
```

```
While(abs(score-prevScore)>0.01)
```

```
For I in ObservedSeq do:
```

```
 Caculate( $\hat{\alpha}$ );caculate( $\hat{\beta}$ );    // update lamda
```

parameters

Caculate (ɔ̃);caculate (ə);

Pi = reCaculate; A = reCaculate; B = reCaculate;

```
PrevScore = score;
```

```
Score = scoreCal();
```

The algorithm discussed above generates a LHMM model for a similar user group. And its time complexity is determined by the iteration times and the number of training samples, so that it is proportional to the k taken.

4.2 Next service and location prediction

After the three important parameters of LHMM are generated, it is time to predict user's future possible behavior. Assuming that $CK=\{O_1, O_2 \dots O_t\}$ is a check-in activity sequence, where O_i represents the check in activity at time i , corresponding to the check-in place sequence $S=\{S_1, S_2 \dots S_t\}$. Now, in order to derive the activity O_{t+1} at time $t+1$, it can analyze the most probable hidden state S_{t+1} at time $t+1$ by applying Equation 6-7 in this paper.

$$p(o_{t+1}|o_{1:t}) = \sum_{s_{t+1}} p(o_{t+1}|s_{t+1}) \sum_{s_t} p(s_{t+1}, s_t|o_{1:t}) \quad (6)$$

$$\sum_{\mathbf{s}_{1:t}} p(\mathbf{s}_{t+1}, \mathbf{s}_t | \mathbf{0}_{1:t}) = \frac{1}{\sum_{\mathbf{s}_{1:t}} p(\mathbf{s}_{1:t})} \sum_{\mathbf{s}_{1:t}} p(\mathbf{s}_{t+1} | \mathbf{s}_t) \alpha(\mathbf{s}_t) \quad (7)$$

Where $\Pr(\mathbf{O}_{1:t} | \mathbf{S}_1)$ represents the occurrence probability of observation O at time $t+1$ under given hidden state $\mathbf{S}_1$. Moreover, $\Pr(\mathbf{S}_t | \mathbf{S}_1)$ shows the state transition probability from time t to time $t+1$. $\Pr(\mathbf{O}_t | \mathbf{S}_t)$ represents the probability of observing $O_{1:t}$ at time t under hidden state S .

Each calculated pair $\langle \text{Si}, \text{Oj} \rangle$ forms an array of service-location probability. By sorting this array, the most probable activity and its corresponding location will be easily worked out with the LHMM which user may take in the next period.

4.3 CF-Behavior prediction algorithm (LHCF)

If it is just stopped at previous module, there exists a huge issue that the recommendation within a group has the same result. In fact, although users share similar routine of the day, it doesn't imply that they enjoy similar interests. For instance, users A and B usually have lunch during 11:00-13:00 at District X. User A prefers restaurant a while user B prefers b .

To solve this problem, a CF-Behavior prediction algorithm (LHCF) based on LHMM is proposed to

generate the final recommended service list for an active user.

when combined with the output generated by previous section, the large matrix can be easily divided into several sub-matrixes at space(S_i) and category(C_i) level.

Fig. 4. Divide dataset into subset by space and category information

According to the above dataset, user's preferences can be described in following two aspects:

- (1) User likes the shop even better when the score is higher
- (2) User likes the shop even better when the visit time is frequent

However, users' interests vary with time. Therefore, a time transfer function is added into final formula, so that users can be clustered in a more proper way. Equation 8-9 are the definition of user's point of interest (POI):

$$POI(U_i, S_j) = \frac{avg(score)}{m \cdot maxscore} + b \cdot \frac{count(s_j)}{\sum_{j=1}^n count(s_j)} \cdot t(U_i, S_j) \quad (8)$$

$$t(U_i, S_j) = 1 - \frac{currentDate - maxDate}{7} \quad (9)$$

Where t function is an interest migration function, the more frequently a user visit a shop, the higher score it will be. Convert the score into number of week can expanding the influence of users' current interest. Attributes a and b are two fit parameters in order to calculate the POI in a more flexible way. Maxscore is a constant number, which depends on how maximum a user can score to a certain shop. Comparing to other users in a certain service, the average score he has rated and the total counts he has visited help the system decide whether this user interests in this category. For check in records without user's rate, an average score will be assigned according to user's historical records.

After generating multiple interest matrixes for different activities and different regions, the cosine similarity calculation formula is applied to calculate the similarity between the mobile users:

$$Sim(U_i, U_j) = \frac{\sum_{k=1}^n POI(U_i, S_k) \cdot POI(U_j, S_k)}{\sqrt{\sum_{k=1}^n POI(U_i, S_k)^2} \cdot \sqrt{\sum_{k=1}^n POI(U_j, S_k)^2}} \quad (10)$$

$POI(U_i, S_k)$ means the points of interest of user i for service k . $POI(U_j)$ presents the average points of interest the user i for all service categories. The higher value of $Sim(U_i, U_j)$, the more similar the two users are.

The pseudo-code of CF-Behavior prediction algorithm is listed as follow:

Algorithm 3. CF-Behavior prediction

Fig 4 shows an illustration of the dimension reduction process.

Input: User: U , prevObservList<location,activity>, HMM parameters

Output: ProbList<location,activity>, RecommendList

Algorithm:

```

pastObserv = buildPastObserv(prevObservList,time) //predict
t+1 service through <O1...Ot>
stateList = getSortedProbHiddenState(pastObserv); //
probability calculation of next state
for statel in stateList do // For each predicted state, calculating
the probability of next observation sequence
observNext<<state,observe>,prob> =
insertAndCalculateProb(pastObserv,statel);
getTop5Prob(observNext); // Probability values are sorted and
selected the top five combinations for <region,activity> in
<state,observe> do // Iteratively predicted next combination
for userU in users do
for CkR in userU.checkinRecords do
if userU.checkinRecord.region = region and
checkinRecord.categories = activity
POI[userU][userU.CkR.shop] = calculatePOI();
UserSim = calculateCosSim(POI) // Calculate the cosine
similarity
topSimUserList[] = topUserSim(userU,5) // Get nearest 5 users
RecommendList<region,activity> =
findHigherPOIShop(topSimUserList[])
// From the similar users group, find out recommendations
users might like shops
    
```

Based on the service sequences which users have participated in as historical records, a numeric probability list is calculated, indicating the likelihood of each possible service and its corresponding occurred places. After the most possible service is determined, the cosine similarity between two users is calculated based on user's historical rating behavior, visit frequency and interest shift. Thus, the final activity recommendation of an active user closely meet his/her personal requirement according to the improved CF-behavior prediction algorithm.

5. EXPERIMENTAL EVALUATION

In this section, the dataset used in our experiment is introduced at first, as well as the efficiency and accuracy of

the two proposed algorithms which have been described in the previous sections.

5.1 Data analysis

Dianping website¹, a famous leading third-party website that enables consumer to review and comment a certain shop, has provided detail business information, consumer reviews and other O2O trading services. Through its open API, our dataset has nearly 6000 shops in Shanghai, along with 600,000 check in records from 3000 distinct users from 2010 to 2014, shown as Figure 5.

Fig. 5. Different services check-in frequency chart

After eliminating the duplicated and abnormal records in the dataset, the experiments are conducted among 100 selected users. The 5-fold cross-validation method is used in this paper. For each one user, the check-ins is divided into 5 subsets. Every time, one of the 5 subsets is used as the test set S_{test} and the other 4 subsets are put together to form training set $S_{training}$. The detail is as follow:

5.2 Behavioral prediction evaluation

The purpose of this experiment is to test prediction accuracy on TopK-LHMM algorithm compared with other traditional methods.

First, the check-in records are grouped by day and labeled as 6 digits, representing district, events and area respectively. Especially, "-" is marked if there is no data in that period. Using the Equation 1, a sequence similarity matrix can be obtained, as shown in Table 1. By the cross-eliminating algorithm, it's easy to calculate the similarity between user A and user B, which is 2 and less than user A and user C, since the result of their similarity calculation is 18.

Table. 1. Behavioral sequence similarity calculation

User Check-In	A:-, -, 100102, 100102, -, -	A:-, -, 100105, -, 100105, 090103
B:-, -, 020104, -, 020204, -	2(phase3)=2	2(phase3)-3(phase5)=-1
B:-, -, -, 020203, -, -	-3(phase4)=-3	0

C:-, -, -, 100110, 130110, -	6(phase4)=6	2(phase5)=2
C:-, -, -, 100203, 100105, 09020	4(phase4)=6	7(phase5)+5(phase6)=12

In order to test the performance of the generated LHMM model, we take one sequence <food, entertainment, hotels> from test set of user A for example. The <food, entertainment> is considered as an observed sequence, and by applying the Equation (6-7), the probability of each service which may be occurred in the third period is generated. According to the experiment, the probability array of user A in the next step is [hotel = 2.43E-3, married = 1.26E-3, dining = 1.03E-3 ...], and the most probable next action is hotel, which is the same as the user's actual behavior, so TRUE is marked for this case test.

To demonstrate the prediction accuracy of the algorithm, the concept of N-hit is introduced. As the example described earlier, the biggest probability value is the one user participates in reality, so this scenario is defined as 1-Hit. However, if user chooses wedding related activity, it is defined as 2-Hit. All in all, N-hit means that the n th value in the prediction array is match with the next step that happens in reality. Obviously, the smaller n is, the higher accuracy the algorithm indicates.

Fig. 6. Experiment on Nearest K value

Fig 6 shows the experiment result of different nearest k value. The performance is very close when k is 5 or 10, but when k is bigger than 10, the prediction of users' next action will drop. The reason of this performance is that the user behavior is similar, not exactly the same. Expanding the number of similar users will reduce the accuracy of the model prediction, but enhance the prediction of coverage. Weighing the pros and cons, we assign k as 10 in follow-up experiments.

As the Figure 7 shows, the accuracy of action prediction for first two hit is over 50%, which has better effect than traditional HMM. If Top K feature adds into LHMM, the new model helps improve additional 5% better accuracy on the 1-hit to 3-hit in average.

¹ <http://www.dianping.com>

Fig. 7. Behavioral prediction comparison

5.3 Interests Recommendation Evaluation

Different from traditional collaborative filtering, LHCF use the output $\langle \text{service, area, probability} \rangle$ from LHMM to divide whole database into several subsets. Check-in records in a certain subset have the same service and district, the similarity between two users is calculated by putting users' score, visit frequency and interest migration into an interest function. After performing the similarity calculation, we can find out which shop may meet the target users' personal demands based on their historical data.

For evaluating the recommendation algorithms, we compare ours with the following five state-of-the-arts methods to see which one provides better recommendations.

- (1) **Location-based Collaborative Filtering (LCF)**: It regards the check-in as item directly and predicts the user ratings of a service with ordinary collaborative filtering algorithm. So that we can use the similar users to do recommendation.
- (2) **User interest and Proximity (UP)**(FERENCE, 2013): Exploring what influence being on users' decision under the space property, it filters out the services that far away from user current location, because of the restriction of human movement distance and the cost of movement.
- (3) **User interest and geographical influences (UG)**(YE, 2011): The user interest components are achieved through the collaborative filtering. The geographical influences is calculated by a power-law probability model, which is used to capture user space-gather activities.
- (4) **Spatio-Temporal Collaborative Filtering (STCF)**(YUAN, 2013): It integrates time information and other factors into the traditional collaborative filtering algorithm. After dividing the time in hours, the STCF model finds a k number of nearest-users that has similar preference in period t to makes a recommendation list for the target user.
- (5) **Location-aware Hidden Markov Model(LHMM)**: Based on the performance of service prediction in LHMM algorithm, it plays a role that predicts which action the target user will choose. Given current observed state O_t and the previous hidden state S_{t-1} ,

corresponding locations S_t can be calculated as Equation 11:

$$\text{Prob}_{S_t} = A_{S_t} \cdot B_{S_t}(O_t) \quad (11)$$

Table 2. Algorithms comparison

	LCF	UP	UG	STCF	LHMM	LHCF
Interest	✓	✓	✓	✓		✓
Space		✓	✓		✓	✓
Time				✓	✓	✓

As discussed above for the experiment, it is the evaluation indicators $\text{hit}@N$ and strategies in papers [16, 17] that we use to evaluate the effectiveness of different recommendation algorithms.

$$\text{hit}@N = \frac{\text{Number of successful recommendations}}{\text{Number of recommendations}} \quad (12)$$

The N represents the number of recommended services. The S_{success} represents the number of success in S_{test} . The S_{test} is a test case set. As for an individual test case $(u, s, l) \in S_{\text{test}}$, the u represents user, s represents service, l represents location. The evaluation procedure is as follows:

- (1) First, we set up test user's current temporal and spatial properties. For example, if a shop is accessed by a user at a coordinate (latitude, longitude) which is (32.56, 23.46) at 12 o'clock. Then, the simulate experiment is at coordinate(32.56+ b , 23.46+ b) at moment (12+ a). The parameters a and b are random and suitable values.
- (2) Then, different algorithm works out its Top- N recommendation list.
- (3) Finally, if a Top- N recommendation list includes the testing service, it is successful and the number of S_{success} increases one..

The $\text{hit}@N$ can reflects the quality of recommendation algorithm. A higher value indicates a better performance to the recommendation algorithm.

Fig. 8. Comparison experiments

Figure 8 shows the performances of the recommended algorithm in each assumptive situation. According to the human custom, we believe that the proper length of the list should be controlled between 1 and 20. The detailed analyses of the experiment results are given as following:

- (1) As for spatial attribute. Compared with LCF, UG and UP algorithms, UG and UP performs better with adding spatial properties. The UG focuses on mining the user's historical location data and has advantage on users' regular daily activities. While the UP focuses on the user's current location context and benefits from that human tend to pick up a close distance service.
- (2) As for time attribute. Compared with LCF and STCF algorithms, STCF benefit from the relative regularity of human daily life, which leads to a narrow range of the recommendation list. Most users choose a restaurant at lunchtime. But time factor in STCF is disordered, so that it can't deal with continuous activities, such as eating first and other activities later.
- (3) As for user interest. Compared with LHMM and LHCF algorithm, both of them use the LHMM model proposed in this paper and perform better than other recommendation algorithms. The difference is that LHCF pay more attention to users' interests. Users may have regular life rhythm, but where the users check in depend on varied factors.

The experiment shows that predictive accuracy of our algorithm is greatly superior to the others in every stage of N . The LHCF algorithm uses LHMM model to predict the next type of service under the current time and space conditions. For example, recommending a cinema to users is usually an invalid recommendation at their lunch time. What's more, ignoring the user's current area, recommending a distant shop to users is almost unsuccessful. The both problem of time and space are solved in our algorithm. In addition, we consider the user's personal interests, so the accuracy of Top-k list can be greatly improved.

6. CONCLUSIONS

In this paper, a user behavior prediction and recommendation framework for location based services is proposed under mobile Internet environment. Based on the users' activity behavior sequences clustering module for location aware mobile services, a Top K-LHMM algorithm is proposed and implemented to do a better prediction for different kind of services and regions under a certain user's status. According to the outputs of predicted services and regions, the dimension of the user's interest matrix is greatly reduced, so that the calculation of the online recommendation module can be more efficient to produce satisfactory services for different users. On the other hand, the recommendation module combines the key factors, including the user's ratings, visit frequency, interest migration, which improves the quality of personalized service recommendation.

Under the extensive experiments designed in this paper, the improved system gives us more accurate results. Especially it overcomes the weakness of perception of location and time context in traditional collaborative

filtering algorithm and obtains a high efficiency on mobile service prediction. The improved system has strong scalability to adapt to different services recommendation environment.

In the future, our research work is mainly focused on how to extend the framework in several directions. First of all, a more precise HMM model can be set up, which is not only taking care of the time and spatial attributes, but also concerning the weather, participants and other situational factors. Then, the overall framework is still too heavy to process in real time environment. To achieve the goal of effectiveness and scalability, the algorithm complexity need to be further optimized in the future. Thus, more experiments need to be taken to verify the improved idea under the location based services environment.

7. ACKNOWLEDGEMENT

This work is partially supported by National Natural Science Foundation of China (61074135, 61303096, 71101086) and Shanghai Leading Academic Discipline Project (J50103). We also would like to show our great appreciations to all of our hard working fellows in the projects above.

8. REFERENCES

- Natarajan N, Shin D. Which app will you use next: collaborative filtering with interactional context[C]. Proceedings of the 7th ACM conference on Recommender systems. (2013) : 201-208.
- Julie D. Thompson, Toby. J. Gibson. Multiple Sequence Alignment Using ClustalW and ClustalX. Current protocols in bioinformatics (2002): UNIT 2.3.
- Blasiak Sam, Huzefa Rangwala. A hidden markov model variant for sequence classification. International Joint Conference on Artificial Intelligence 22(1)(2011):1192-1197.
- Antwarg, Liat, Lior Rokach. Attribute-driven hidden markov model trees for intention prediction. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 42(6)(2012): 1103-1119.
- Hamada R, Kubo T, Ikeda K. Towards prediction of driving behavior via basic pattern discovery with BP-AR-HMM. Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference 2013 : 2805-2809.
- Mathew, Wesley, Ruben Raposo. Predicting future locations with hidden markov models. Proceedings of the ACM Conference on Ubiquitous Computing 2012: 911-918.
- Daniel Lemire. Slope one predictors for online rating-based collaborative filtering[J]. Society for Industrial Mathematics(2005).
- Yuan, Guan, Shixiong Xia. Interesting activities discovery for moving objects based on collaborative filtering. Mathematical Problems in Engineering 2013.
- Jihang Ye, Zhu Zhe. What's your next move: User activity prediction in location-based social networks. Proceedings of the SIAM International Conference on Data Mining (SDM) 2013: 171-179
- Zukerman, Ingrid, David W. Albrecht. Predicting users' requests on the WWW [B]. Springer Vienna (1999)
- Welch, Lloyd R. "Hidden Markov models and the Baum-Welch algorithm." IEEE Information Theory Society Newsletter 53.4 (2003): 10-13.

Ference G, Ye M, Lee W C. Location recommendation for out-of-town users in location-based social networks[C]. Acm International Conference on Conference on Information & Knowledge Management. ACM, 2013:721-726.

Ye M, Yin P, Lee W C, et al. Exploiting geographical influence for collaborative point-of-interest recommendation[C]. Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. ACM, 2011:325-334.

Yuan Q, Cong G, Ma Z, et al. Time-aware point-of-interest recommendation[C]. Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. ACM, 2013:363-372.

Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, Usa, August. 2008:426-434.

Yin H, Cui B, Li J, et al. Challenging the Long Tail Recommendation[J]. Proceedings of the Vldb Endowment, 2012, 5(9):896-907.

Authors

Mingjun Xin is a full professor in the School of Computer Engineering and Science, Shanghai University. He received his PH.D. from School of Computer Science and Technology, Northwestern Polytechnical University, Xi'an China in 2001. His research

interests are Quality of Service (QoS), Location-Based Services, mobile platform, service recommendation and software engineering.

Shunxiang Li is a M.S. student in School of Computer Engineering and Science, Shanghai University. He received his B.S. degree from School of Computer Science, Taiyuan University of Technology, China in 2014. His research interests are recommendation system, mobile

platforms.

Liyuan Zhou is a M.S. student in School of Computer Engineering and Science, Shanghai University. She received her B.S. degree from School of Computer Science, Shanghai University, China in 2013. Her research interests are recommendation system, mobile

platforms.

Weimin Li is a associate professor in the School of Computer Engineering and Science, Shanghai University. His research interests are social network analysis, cloud data management, social computing, optical computing applications.