Exploiting Space-Time Status for Service Recommendation

Changjian Fang, Bo Mao, Jie Cao, and Zhiang Wu*

Jiangsu Provincial Key Laboratory of E-Business,
Nanjing University of Finance and Economics, Nanjing, P.R. China
zawuster@gmail.com

Abstract. The prevalence of smart devices allows people to record their space-time status. This paper focuses on exploiting user space-time status and the related semantic information for service recommendation. Firstly, event DAG is employed to organize the space-time information generated based on the service invocation history. Generation algorithm of the event DAG is then proposed. Secondly, a novel collaborative filtering based recommendation algorithm is designed. Potentially interesting services in the target node and its subsequent nodes can be recommended. In our implementation, the user space-time status is generated from the 4D city models (3D location + time) with semantic information. A prototype system is implemented to generate service invocation logs of different users. These simulative logs are utilized to evaluate the effectiveness and efficiency of our proposed method.

Keywords: 4D city model, space-time status, service recommendation.

1 Introduction

Along with the development of wireless networking, portable mobile devices and mobile broadband Internet access technologies [1,2], it becomes convenient to acquire space-time status of the mobile user. It is possible for the user to enjoy recommendation of services at any time in any place. The requirements of the people are related with their location and time. Meanwhile, the prevalence of Web 2.0 technologies leads to a great increscent of real time generated data [3]. It is getting more difficult to make a recommendation for a user because of not only huge amount data, but also lack of semantic information of the user.

This paper focuses on exploiting space-time status and related semantic information of users for service recommendation. The location of the user can be obtained by the GPS or mobile positioning method. The semantic information of location can be gained from the semantic city models. The OGC 3D city model standard [4], CityGML, defines the semantic representation of urban objects e.g. the usage of a building. Based on the position and semantic 3D city models, we can identify the semantic location of the user e.g. a shopping mall, a traffic station or a hospital. By combining

* Corresponding author.

H. Wang et al. (Eds.): APWeb 2012 Workshops, LNCS 7234, pp. 245–256, 2012.
© Springer-Verlag Berlin Heidelberg 2012

3D location and time (or 4D city model), more semantic information will be revealed such as local holiday, weather condition, big events, and so on. In this paper, the concept of service is not limited to Web Service, but the activity from which people can reap the benefits e.g. alarm clock, news, traffic, discounting, and so on.

It is possible and necessary to recommend the services for the users according to their positions and the current time. In fact, the requirements are not limited to the 3D location and time but their semantic information behind. In this paper, we try to find out the relationship between user requirements and their space-time status information and make use of that for the service recommendation.

The rest of the paper is structured as follows. Related work is given in section 2. Section 3 describes the definition and generalization of user space-time status. Section 4 gives the service recommendation mechanism based on the proposed space-time status. A stimulated case study is shown in section 5. Section 6 concludes the whole paper.

2 Related Work

2.1 Recommender Systems

In recent years, recommender systems have emerged as one tool helping people look for items that they are interested in. The items include commodity products, movies, advertisements, CD, Web services, and so forth. The recommendation algorithm, the kernel of recommender systems, has been a hot research topic for a long time. The initial algorithm used by recommender systems is the content-based algorithm which suggests items that are similar to the ones the current users has shown a preference for in the past [5]. Content-based algorithm relies on rich content descriptions of the items (services for example) that are being recommended. For instance, if a recommender system wants to suggest a coat, the content-based algorithm will depend on information such as brand, style, color, price, etc. Therefore, the designer of the content-based algorithm should obtain abundant domain knowledge which may not be readily available or straightforward to maintain.

Collaborative filtering (CF) is different from the content-based methods. It collects opinions from users in the form of ratings on items. Its advantage over the content-based algorithms is that the CF algorithm does not need a representation of the items but only rely on historical information of users. CF algorithms can be divided into two categories: neighborhood-based approach and latent factor models [6]. User-based CF (UCF), a representative neighborhood-based CF algorithm, is adopted by a multitude of well-known recommender systems such as Tapestry, MovieLens, Amazon Book Store, YouTube, Facebook, and so forth. UCF utilizes the opinions from a user's *k* nearest neighbors (*k*NN) to help the user to identify his/her interested content from an overwhelming set of potential choices. In this paper, we propose a new CF algorithm in order to incorporate semantic 4D (space-time status) information for improving the recommendation.

Adomavicius et, al. use a multi-dimensional approach to incorporate contextual information [7]. Zheng et, al. report on a personalized friend and location recommender for the geographical information systems (GIS) on the Web [8,9]. GPS trajectories in [8,9] are collected by 75 volunteers over a period of one year. Limited by time and condition, the experimental data utilized in this paper is generated by a simulator. However, smart devices will be equipped on volunteers in the future in order to generate genuine dataset including space-time status of users and service invocation records.

2.2 Context Generation

Context aware computation is an interesting subject for many researchers [10-12]. Context is the information/data for characterizing the situation of an entity e.g. a person, place, or object that is considered relevant to the interaction between a user and an application, including location, time, activities, and the preferences of each entity [13,14]. The existing context information employed in the service recommendation is mainly about the user [15]. Along with the development of mobile techniques, the location based context information can be gathered for recommendation [16]. Quercia et al. [17] use location data from mobile phone to recommend the social events.

By reference to the 3D city models, not only location but also the semantic information of the location can be retrieved. Kolbe [18] suggests the semantic information should be contained in the 3D city models, and OGC issued the 3D city model representation standard CityGML [4]. Many cities such as Berlin, Stuttgart have delivered their official 3D city models online in CityGML. We can expect the semantic 3D city model will be available as the city information infrastructure. Through the semantic 3D city models, we can further get the semantic information of a position.

Based on the semantic information of the 3D city models and the time of user service invocation, we can get the space-time status about the service which is essential in the user service recommendation, because most of people have the experience that the need certain service when they are in certain place at certain time. Therefore, the space-time status of user will be generated and stored for the service recommendation in this paper.

3 Space-Time Status

3.1 Definition

We use directed acyclic graph (DAG) to organize the space-time information generated during the service invocation. The DAG is defined as follows:

Definition 1 (Event DAG, G): A DAG $G=<N, E>$ consists of a set of nodes N representing the events and a set of directed edges E representing dependencies among events. Assume there are n nodes and m edges, namely $N=\{node_i|0<i<n\}$and $E=\{edge_j|0<j<m\}$. A event node $node_i=<Time, Location, Services>$ consists of time, location and invoked services. The edge set E contains edges $<node_a, node_b, ber>\in E$ for each event $node_a$ (parent) that $node_b$ (child) depends on.

In fact, a node represents an event: the services invoked in a space-time coordinate (time, location). The edge is the number of the conversion from $node_a$ to $node_b$.

The time is an interval containing start and end. In this paper, the time is looped by the week that most people apply to. In the city area, the surrounding city objects affect the selection of services. For example, we trend to check in to the nearby hotel when the time is late; or find out the closest public traffic facilities. The location is defined as follows:

Definition 2 (Location): The $Location=\{City_object_i | 0<i<nc\}$is composed by the surrounding city objects. A city object is denoted by City_object_i = {Type, Class, Function, Usage} where Type = {Building | Transportation | WaterBody | Vegetation | City furniture} and *Class*, *Function, Usage* are specified in the CityGML standard.

The definitions of the *Class*, *Function* and *Usage* have been specified in the CityGML standard. Generally, each city object has the attributes *class*, function and *usage*, unless it is stated otherwise. The class attribute can occur only once, while the attributes usage and function can be used multiple times. The class attribute describes the classification of the objects, e.g. road, track, railway, or square. The attribute function contains the purpose of the object, like national highway or county road, while the attribute usage may define if an object is e.g. navigable or usable for pedestrians (OGC, 2008). For example *BuildingType* can be habitation, sanitation, sport, education, traffic; *BuildingFunction* can be office building, court, post office.

The services of a node are the user's invoked services during the time interval at a certain location. The semantic 4D (3-dimensional location + time) and the history of the services become the basis of the recommendation.

3.2 Generation

To generate the DAG of user status, we need the service invocation log and the city object database. Initially, the DAG only contains two nodes: the start node and the end node as shown in Fig. 1(a). We then read the service invocation log one by one and construct or modify the DAG accordingly as in Fig. 1(b).

In our implementation, the user log is formatted as: $L=<ID, TS, P, S>$ in which *ID* is the user identity, *TS* is the time stamp, *P* is the coordinate of the user represented by (x, y, z) and *S* is the invoked services. The user log can be automated gathered from the "smart devices" with GPS function. In the specific recommend system, the *S* should be filtered to preserve the interested target services and remove these uninterested ones.

Another resource required for the user DAG creation is the city object database. It should provide the function of objective inquiring from coordinate. The stored city objective not only contains the general semantic information such as name, function, type etc. but also the personal semantic information. For example, one building may be the home for a person and also the work place for another people. Also the time can be decided the personal semantic, such as a same road in different time may be the way home or way work in different time. Therefore, the inquiring is processed in two steps: the first is getting city object (building, road etc) from the coordinate; the second is getting personal semantic information. The semantic information will be recorded in the nodes of user space-time status which are the many bases to implement the recommendation.

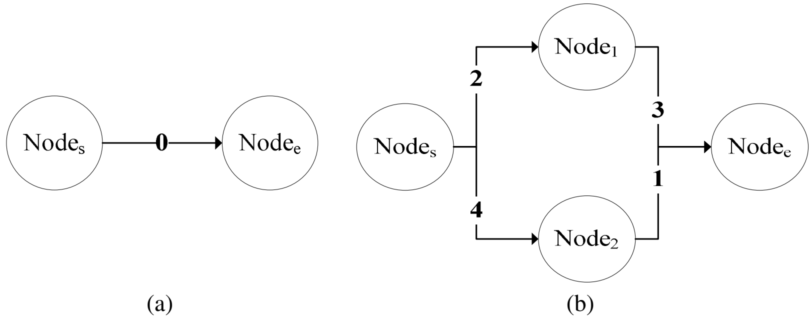


Fig. 1. An example of the space-time DAG

From the user log, DAG of user space-time status is generated as following three steps:

Step 1: Initialization Initially, the DAG of user space-time status just contains two controlled nodes start and end without real semantic information. We define the current node (CN) indicating the present status of the user. The current node is point to the start node in the beginning.

Step 2: Updating In the updating step, fetch one record in the user log as current record. Assume the current record L_i=<ID_i, TS_i, P_i, S_i>, and the current status is $node_j$=<$Time_j$, $Location_j$, $Service_j$>. Then the updating process can be implemented according to the pseudo-code in table 1. Since the time is cycled in the week basis, the current status will automatically go to the start point when the week is over.

Step 3: Simplification Usually the generated user space-time status DAG may contain many nodes and edges when the user log data is large. Therefore, it is necessary to simplify the DAG for faster matching. In the proposed framework, we detect the number of service invoking recorded in edge. If the number is smaller than a set value and one of the nodes in the edge is one degree node, the edge and its connected degree one node will be removed.

Table 1. Pseudo-code of the user space-time status DAG generation

```
Input:   Lset: user service invocation log set
Output:  G=<N, E>: a DAG to describe the user space-time status
1: Initiate the G=<N,E> in which N={Node_s, Node_e} E={<Node_s, Node_e, 0>};
2: Current Node CN=Node_s;
3: for all S_i ∈ Target Service Set
4:     Location_i = getLocation(ID_i, TS_i, P_i);
5:     if location_i is the same as the location of current node
6:         extend time of CN to TS_i; add S_i into the services of CN
```

```
      continue;
  else
      for every node_l in edge_l <CN, node_l, n>
         if location_i is the same with location in node_l
             extend time of node_l to TS_i; add S_i into the services of node_l;
             Edge_l.n+=1; set CN to node_l;
             continue;
         end if
      end for
  end if
  node_k=<TS_i, Location_i, S_i>; edge_k=<node_j,node_k,1>;
  add node_k into N; add edge_k into E;
end for
remove the edge with its n value smaller than 5 and the degree of one of its node is 1.
```

4 Recommendation Based on Space-Time Status

Service recommendation in this paper aims to provide potentially interesting services when the user is in a *target node*. Furthermore, potentially interesting services when the user is the next possible nodes will be also recommended. For instance, assuming the target node of a child is <(7:30, 8:00), home> of which the next possible nodes include <(8:30, 11:30), school>, <(9:00, 12:00), carnie>, or <(8:20, 9:20), bookstore>, all potentially interesting services in these four nodes will be recommended by the proposed approach. First, *k*NN are selected based on the distance computation. Second, the list of recommended services in the target node is generated. Third, the lists of recommended services in subsequent nodes are generated.

4.1 Distance Computation among Nodes and *k*NN Selection

The distance between two nodes in G is measured by the semantic 4D information including time, coordinate, semantic information of city object. The variable types of these data are heterogeneous. For instance, time and coordinate are interval-scale variables, and type, class and function are categorical variables. We propose to utilize Eq. (1) to compute the distance between $node_a$ and $node_b$.

$$distance(node_a, node_b) = \frac{\sum_{f=1}^{p} \delta_{ab}^{f} d_{ab}^{f}}{\sum_{f=1}^{p} \delta_{ab}^{f}} \quad (1)$$

Assume there are p variables of the semantic 4D information. In Eq. (1), if $node_a$ or $node_b$ do not assign a value to the f-th variable, $\delta^f_{ab}=0$; otherwise $\delta^f_{ab}=1$. The distance of the f-th variable between $node_a$ and $node_b$ is written d^f_{ab} which depends on the variable type. We divide the computation of δ^f_{ab} into two cases according to variable types.

- Case (1). the f-th variable is binary variable or categorical variable: if $x_{af} = x_{bf}$, $d^f ab = 0$; otherwise $d^f ab = 1$.
- Case (2). the f-th variable is interval-scale variable: time and coordinate belong to this type. *Manhattan* distance is employed to measure the distance of this type. Let $<x_i, y_i>$ and $<x_j, y_j>$ denote coordinates or time interval of $node_a$ and $node_b$ respectively, and $d^f ab$ can be computed as Eq. (2) shown.

$$d^f_{ab} = |x_i - x_j| + |y_i - y_j| \tag{2}$$

Selecting kNN of nodes is an important step for making accurate recommendation. Let $node_t$ denote the target node and kNN($node_t$) denote k nearest neighbors of $node_t$. Based on the distance calculated by Eq. (1), we select k nodes whose distance with the target node are as small as possible.

4.2 Recommendation in the Target Node

This phase predicts the potentially interesting services in the target node on the basis of its top-k neighbors. Since *Services* in $node_i$ is a binary vector where $S_i=1$ denotes the user has invoked S_i, we predict the interesting degree according to the number of votes of kNN($node_t$). Let $S_{t,j}$ denote predicted interesting degree of S_j in $node_t$, and $S_{t,j}$ is given by Eq. (3).

$$S_{t,j} = \sum_{node_k \in k\mathrm{NN}(node_t)} \frac{S_{k,j}}{distance(node_t, node_k)} \tag{3}$$

where $S_{k,j}$ is the binary value of S_j in $node_k$, and $distance(node_t, node_k)$ represents the distance between $node_i$ and $node_k$ which is calculated as mentioned in Eq. (1). Then, top-N services having the most votes are recommended as recommended services in the target node. Let $RST_N(node_t)$ denote the list of recommended services in the target node, and $RST_N(node_t)$ is given by Definition 3.

Definition 3 (Top-N recommended services, $RST_N(node_t)$). $RST_N(node_t)$ is a set of services with the size N and satisfies that $\forall S_j \in RST_N(node_t),\ \forall S_m \in \overline{RST_N(node_t)},\ S_{t,j} \geq S_{t,m}$, where $\overline{RST_N(node_t)}$ is the complementary set of $RST_N(node_t)$.

Once we have calculated predicted interesting degrees for all services in $node_t$, the list of recommended services with size N can be readily obtained.

4.3 Recommendation in Subsequent Nodes

The target node has several subsequent nodes. The edge between two nodes represents the number of the conversion from which the probability of this conversion can be calculated. Let $node_t^m$ be the m-step node from the $node_i$, and $< node_t^0, node_t^1, \ldots, node_t^m >$ be the path from $node_t$ ($node_t = node_t^0$) to $node_i^t$. Let

$P(node_t^q, node_t^{q+1})$ denote the probability between $node_t^q$ and $node_t^{q+1}$, and $S_{t,j}^t$ denote predicted interesting degree of S_j in the t-step node from the $node_i$. $S_{i,j}^t$ can be computed by Eq. (4).

$$S_{t,j}^m = S_{m,j} \cdot \prod_{q=0}^m P\left(node_t^q, node_t^{q+1}\right) \tag{4}$$

where $S_{m,j}$ represents predicted interesting degree of S_j in $node_t^m$, and $S_{m,j}$ can be calculated by Eq. (3). As the increase of m, $S_{t,j}^m$ verges on a minimum value and the strength of $S_{t,j}^m$ for service recommendation reduces sharply. It is in accordance with the fact that the longer distance between two nodes is, the lower accuracy of prediction is. From the Definition 3 and $S_{t,j}^m$, top-N services in $node_t^m$ can be obtained.

5 Case Study

Since there is no open dataset including GPS trajectories and service invocation histories, we design and implement a prototype system in order to generate service invocation logs of different types of users. We then generate event DAG based on these logs and recommendation in nodes of DAG. In this section, we first present the setup of our prototype system, including user types, events, services and places. Then, we report the experiments of event DAG generation and recommendation based on space-time status.

5.1 Setup

The prototype system simulates behaviors of different types of users on workday or weekend. The user behavior in one day consists of a series of random events in which service invocation histories are kept. Table 2 depicts user types, events, services and places of the prototype system.

Table 2. Attributes and possible values of our prototype system

Attribute	Possible Values
user types	young man	young woman	old man	old woman	child
events	get up	eat breakfast	trip	working	eat lunch	shopping	entertainment
services	alarm_clock	news	traffic	Internet	print	repast	group_purchase
places	home	office	school	road	bazaar	restaurant	bookstore	carnie	resort

Service invocation logs of users are generated in two scenarios that are workday and weekend respectively. For example, a young woman might prefer to read world news in the workday morning, and on weekends to read movie reviews and do shopping.

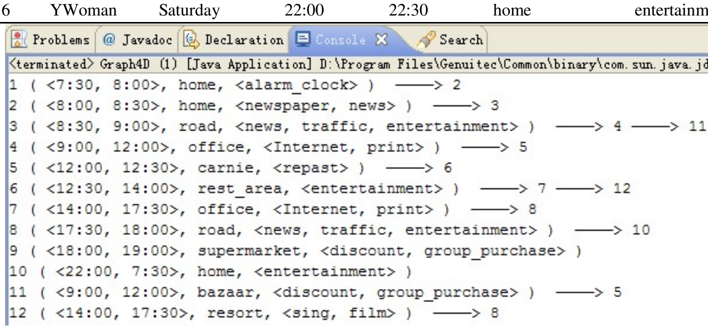
5.2 Experimental Results

5.2.1 Event DAG Generation.

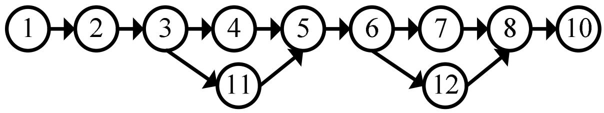
Event DAG generation algorithm shown in Table 1 is implemented in our prototype system. In this section, we utilize an example to reveal the process of the event DAG generation. Table 3 gives a representative log of a young woman on Wednesday and Saturday. *Adjacency list* is employed to store the event DAG in our implementation. Fig. 2(a) gives the output of the adjacency list obtained from the log in Table 3, and Fig. 2(b) shows the corresponding DAG of Fig. 2(a).

Table 3. An example of user log

EventID	UID	UType	Day	Start-time	End-time	Location	Invoked Services
1	6	YWoman	Wednesday	7:30	8:00	home	alarm_clock
2	6	YWoman	Wednesday	8:00	8:30	home	newspaper, news
3	6	YWoman	Wednesday	8:30	9:00	road	news, traffic, entertainment
4	6	YWoman	Wednesday	9:00	12:00	office	Internet, print
5	6	YWoman	Wednesday	12:00	12: 30	carnie	repast
6	6	YWoman	Wednesday	12:30	14: 00	rest_area	entertainment
7	6	YWoman	Wednesday	14:00	17:30	office	Internet, print
8	6	YWoman	Wednesday	17:30	18:00	road	news, traffic, entertainment
10	6	YWoman	Wednesday	22:00	7:30	home	entertainment
1	6	YWoman	Saturday	7:30	8:00	home	alarm_clock
2	6	YWoman	Saturday	8:00	8:30	home	newspaper, news
3	6	YWoman	Saturday	8:30	9:00	road	news, traffic, entertainment
11	6	YWoman	Saturday	9:00	12:00	bazaar	discount, group_purchase
5	6	YWoman	Saturday	12:00	12: 30	carnie	repast
6	6	YWoman	Saturday	12:30	14: 00	rest_area	entertainment
12	6	YWoman	Saturday	14:00	17:30	resort	sing, film
8	6	YWoman	Saturday	17:30	18:00	road	news, traffic, entertainment
10	6	YWoman	Saturday	22:00	22:30	home	entertainment


```
Problems  @ Javadoc  Declaration  Console  Search
<terminated> Graph4D (1) [Java Application] D:\Program Files\Genuitec\Common\binary\com.sun.java.jdl
1 ( <7:30, 8:00>, home, <alarm_clock> )  ——> 2
2 ( <8:00, 8:30>, home, <newspaper, news> )  ——> 3
3 ( <8:30, 9:00>, road, <news, traffic, entertainment> )  ——> 4 ——> 11
4 ( <9:00, 12:00>, office, <Internet, print> )  ——> 5
5 ( <12:00, 12:30>, carnie, <repast> )  ——> 6
6 ( <12:30, 14:00>, rest_area, <entertainment> )  ——> 7 ——> 12
7 ( <14:00, 17:30>, office, <Internet, print> )  ——> 8
8 ( <17:30, 18:00>, road, <news, traffic, entertainment> )  ——> 10
9 ( <18:00, 19:00>, supermarket, <discount, group_purchase> )
10 ( <22:00, 7:30>, home, <entertainment> )
11 ( <9:00, 12:00>, bazaar, <discount, group_purchase> )  ——> 5
12 ( <14:00, 17:30>, resort, <sing, film> )  ——> 8
```

(a)



(b)

Fig. 2. The adjacency list and its corresponding DAG

5.2.2 Performance of the Recommendation

Service invocation logs of different types users in one week are generated firstly. An event DAG is constructed for one user in a week. We then select some target nodes and some of their subsequent nodes, and suppose the invoked services in these selected nodes are unknown. The list of recommended services are generated by the method mentioned in Section 4. Performance of the recommendation method is evaluated by comparing the list with the logs. The metric of *Precision* is defined as follows:

$$Precision = \frac{N_{rs}}{N_s} \tag{5}$$

In Eqs. (5), N_{rs} is the number of services in both recommended list and the list of invoked services recorded in the log, and N_s is the length of the recommended list.

The first experiment investigates how m affect the accuracy. We range m from 0 to 6 and set $k=8$ and $N=5$. We select 10 target nodes in the event DAG of five types of users on workday and weekend respectively. Fig. 3 shows how the precision of recommendation vary with the increase of m. As m increases, the precision decreases firstly and then varies irregularly. The smaller distance from the target node is, the higher precision is. However, when m reaches a bigger value, the stringency of recommendation reduces sharply, resulting in the irregularity of the precision variance.

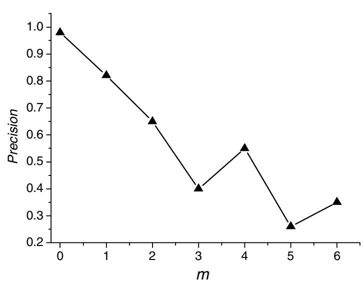


Fig. 3. Impact of m on precision of recommendation

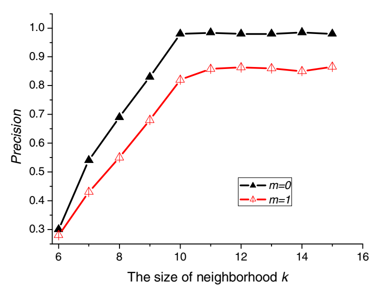


Fig. 4. Impact of k on precision of recommendation

The second experiment investigates how k affect the accuracy. We select 10 target nodes in the same way. And we range k from 6 to 15 and set $N=5$. Recommendation results on the target nodes and their single-step nodes are recorded. The experimental results shown in Fig. 4 indicate that as the increase of k, precision in two scenarios increases firstly and then tends towards stability. Therefore, it is not necessary to select nearest neighborhoods as much as possible.

6 Conclusion

This paper focuses on exploiting space-time status and its related semantic information of users for service recommendation. We propose to utilize event DAG to organize the space-time information generated during the service invocation. The generation algorithm of the event DAG is also proposed. Based on the event DAG, a novel collaborative filtering based recommendation algorithm is presented. Potentially interesting services in the target node and its subsequent nodes can be recommended. A prototype system is designed and implemented to generate service invocation logs of different users. We then utilize an example to elaborate the process of the event DAG generation. Finally, how parameters m and k affect the precision of recommendation is demonstrated by experimental results.

In the future, smart devices will be purchased and volunteers will be recruited so that abundant realistic service invocation logs with 4D city model can be obtained. We will propose more skillful recommendation algorithms and conduct more detailed experiments.

Acknowledgement. This research is supported by National Natural Science Foundation of China under Grants Nos.71072172 and 61103229, Industry Projects in the Jiangsu S&T Pillar Program under Grants No.BE2011198, Jiangsu Provincial Key Laboratory of Network and Information Security under Grants No. BM2003201, Transformation Fund for Agricultural S&T Achievements under Grants No. 2011GB2C100024 and Innovation Fund for Agricultural S&T in Jiangsu under Grants No. CX(11)3039.

References

1. Wang, S., Wu, C.: Application of Context-aware and Personalized Recommendation to Implement an Adaptive Ubiquitous Learning System. Expert Systems with Applications 38(9), 10831–10838 (2011)
2. Cao, J., Wu, Z.: An Improved Protocol for Deadlock and Livelock Avoidance Resource Co-allocation in Network Computing. World Wide Web: Internet and Web Information Systems 13(3), 373–388 (2010)
3. Wu, D., Ke, Y., Yu, J., Yu, P., Chen, L.: Leadership Discovery when Data Correlatively Evolve. World Wide Web: Internet and Web Information Systems 14(1), 1–25 (2011)
4. Gröger, G., Kolbe, T.H., Czerwinski, A., Nagel, C.: OpenGIS City Geography Markup Language (CityGML) Encoding Standard Copyright. Open Geospatial Consortium Inc. (2008)
5. Melville, P., Mooney, R., Nagarajan, R.: Content-boosted Collaborative Filtering for Improved Recommendations. In: 8th National Conference on Artificial intelligence (AAAI 2002), pp. 187–192 (2002)

6. Koren, Y.: Factor in the Neighbors: Scalable and Accurate Collaborative Filtering. ACM Transactions on Knowledge Discovery from Data 4(1), 1–24 (2009)
7. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach. ACM Trans. on Information Systems 23(1), 103–145 (2005)
8. Zheng, Y., Zhang, L., Ma, Z., Xie, X., Ma, W.: Recommending Friends and Locations based on Individual Location History. ACM Trans. on the Web 5(1), Article 5, pages 44 (2011)
9. Zheng, Y., Xie, X.: Learning Travel Recommendation from User-generated GPS Traces. ACM Trans. on Intelligent Systems and Technology 2(1), Article 2, pages 29 (2011)
10. Brown, P.J.: The Stick-e Document: a Framework for Creating Context-aware Applications. In: Proceedings of EP 1996, Palo Alto, pp. 259–272 (1996)
11. Rosemann, M., Recker, J.: Context-aware Process Design: Exploring the Extrinsic Drivers for Process Flexibility. In: 18th International Conference on Advanced Information Systems Engineering, Proceedings of Workshops and Doctoral Consortium, pp. 149–158 (2006)
12. Schmidt, A., Beigl, M., Gellersen, H.: There is more to Context than Location. Computers & Graphics Journal 23(6), 893–902 (1999)
13. Abowd, G.D., Atkeson, C.G., Hong, J., Long, S., Kooper, R., Pinkerton, M.: Cyberguide: a Mobile Context-aware Tour Guide. Wireless Networks 3(5), 421–433 (1997)
14. Yau, S.S., Karim, F.: An Adaptive Middleware for Context-Sensitive Communications for Real-Time Applications in Ubiquitous Computing Environments. Real-Time Systems 26(1), 29–61 (2004)
15. Adomavicius, G., Tuzhilin, A.: Context-aware Recommender Systems. In: Proceedings of the 2008 ACM Conference on Recommender Systems (RecSys 2008), pp. 335–336 (2008)
16. Park, M.H., Hong, J.H., Cho, S.B.: Location-Based Recommendation System Using Bayesian User's Preference Model in Mobile Devices. In: Indulska, J., Ma, J., Yang, L.T., Ungerer, T., Cao, J. (eds.) UIC 2007. LNCS, vol. 4611, pp. 1130–1139. Springer, Heidelberg (2007)
17. Quercia, D., Lathia, N., Calabrese, F., Lorenzo, G.D., Crowcroft, J.: Recommending Social Events from Mobile Phone Location Data. In: IEEE ICDM, Sydney, Australia (2010)
18. Kolbe, T.H.: CityGML–3D Geospatial and Semantic Modeling of Urban Structures. Presentation on the GITA/OGC Emerging Technologies Summit in Washington, http://www.citygml.org/fileadmin/citygml/docs/CityGML_ETS4_2007-03-21.pdf (accessed July 2, 2011)