A Journey Recommender System Using Crowd Attention Monitoring for Facilitating a Collaborative Visiting Experience

Rebecca Chen, Tina Ho and Raymund J. Lin
Software Development Lab
International Business Machines Corporation
Taipei, 115, Taiwan

ABSTRACT

Nowadays when people are visiting an exhibition or museum, the only available guidance is usually brochure or map, which shows location information statically. People are not able to get real-time situation of the exhibition or museum, such as which booth is currently popular. With the prevalence of mobile and social network, we can analyze people's booth visiting histories and provide valuable insights to users of a proposed journey recommender system. This presentation proposes a journey recommender system using crowd attention monitoring and dynamic route planning for facilitating a collaborative visiting experience. An exhibition case is used to demonstrate how the system works.

Keywords: Collaboration, Social, Recommendation System

INTRODUCTION

Nowadays, we can easily see recommendation functions in online systems, such as on-line shopping, article reading, or map navigation services. In our real life, we might also expect to use such a recommendation system to guide us during the tour of World Expo, theme park, museum or exhibition, instead of reading a brochure or paper map on hand. How to build up a journey recommender system using crowd attention monitoring for facilitating a collaborative visiting experience will be the essential to implement such a recommendation system. In this paper, we present a crowd-sourcing recommender system that provides dynamic recommendation to the visitors by finding similar visitors in the system based on his/her behavior/characteristics. The recommendations are generated dynamically and customized for each individual, based on his/her attention "budget" with regarding to those who (1) has the similar attention pattern (2) has the similar interests, or profile, i.e., this matching mechanism leverages dynamic and real-time trends to provide the most suitable arrangement for the visitors. Besides, we consider the congestion condition of the venues, so the attention budget could be calculated correctly. Thus, this recommender system as a service could be used to largely improve user experiences in exhibition, museum, or similar events.

This paper is organized as follows. Section II discusses some existing recommendation systems. Section III introduces the architecture of our crowd-sourcing recommender system, and Section IV gives an example utilizing such a recommender system. Finally, we conclude this paper in Section V.

RELATED WORK

Content Based Filtering

Content Based Recommendation system is a traditional recommender that provides recommendations based on its context or users' frequent behaviors. For example, if a person who goes to Amazon book store and buys some books about jogging, the content based recommender will recommend some other jogging/exercise related book to this person. If a person who usually buys the Disneyland ticket promotion in spring sale, then whenever there is a spring promotion sales for Disneyland, the recommendation system automatically provide this recommendation to him/her when the promotion starts. As mentioned in the study (Balabanoviæ, M., & Shoham, Y., 1997), the content based recommendation system has many disadvantages. Only shallow analysis of certain kinds of content can be supplied.

Collaborative Filtering

Collaborative Recommendation system provides recommendations for a user based solely on the basis of similarities between users. The collaborative recommendation system is common in our daily on-line shopping experience. For example, the system observes that most of people who buy the goggle also buy the earplug and sunscreen. Therefore, if there is a user buying goggle, the recommender will ask the user if he/she is interested in buying earplug or sunscreen. Pure collaborative recommendation addresses the shortcomings given pure content-based systems. However, this approach introduces new problems of its own. For example, if there is any new item in the database, the collaborative filtering will not have any recommendation to provide. Or if a user whose taste is unusual compared to the majority, the recommendation provided might not be useful to him/her.

Hybrid Content-Based and Collaborative Recommendation

CrowdPlanner (Su, 2013), a crowd-based route recommendation system, is a typical instance of a collaborative filtering recommendation system. It requests human workers to evaluate candidates routes recommended by different sources and methods, and determine the best route based on the feedbacks of these workers. That means, it leverages crowds' knowledge to improve the quality of recommendations. The cognition of routes of humans is quite different from computers' that human recognize routes by preferences and experiences of the real situation while computer recognizes routes from continuous time and data recorded. Thus, to provide the recommendations by evaluating the best route with user as one of the collaborative filtering is considered better. Oracle white paper (Oracle White Paper, 2011, March) endorsed such a hybrid content-based recommendation system.

ARCHITECTURE OF CROWD-SOURCING RECOMMENDER SYSTEM

In this section, we introduce the architecture of dynamic customized visiting journey recommendation system and its components. We also illustrate how the recommendation system works. Figure 1 presents the proposed dynamic customized visiting journey recommendation system. It consists of six key components:

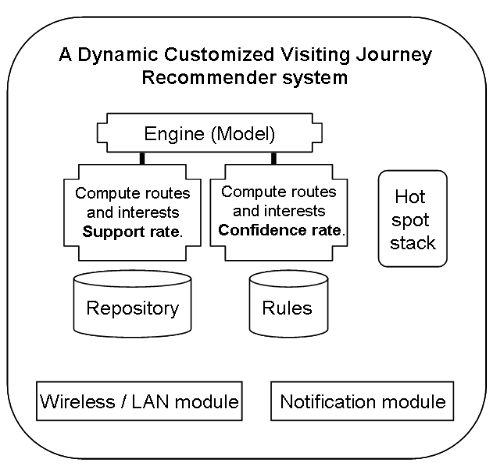

Figure 1 Components of the dynamic customized visiting journey recommendation system

Filtering Engine (Core Model)

The engine is the core component of our recommendation system. It is responsible for the following tasks:

- Pick up history records from repository and calculate each record’s supporting rate.
- Pick up rules from rule repository and calculate each rule’s confidence rate.
- Retrieve/store rules from/to repository.
- Compute congestion ratio of each location.
- Provide recommendation by sorting and filtering the rules according to user’s preference.
- Push the recommendation to hot-spot stack when congestion occurs. Pop up recommendation when congestion situation is released.

Rules

The rules repository is designed to store the recommended rules and each rule is calculated with a confidence rate.

Repository

This repository is designed to store the locations (ex. booths) that users have visited and the attraction time of that location to the user. Each history record is saved in repository and calculated with supporting rate.

Hot Spot Stack

The hot spot stack is a repository used for store the recommendation when congestion ratio passed a threshold. If the congestion ratio of one booth is less than the threshold, system may notify user he/she can go to that booth he/she is interested in before.

Notification Module

The notification module is the module that pushes the notification messages to user’s device.

Wireless / LAN Module

The Wireless/LAN module is the module handling all network connections from/to our recommendation system.

Based on the architecture, the recommendation system provides the recommendation to a visitor by finding a similar visitor in the system based on his/her behavior/characteristics. The recommendation will be generated dynamically and customized to each individual. Our system also considers the visitor's personal preference such as how much time he/she wants to spend in the trade show? What is their interested area? How many booths they plan to visit during the trade show? Figure 2 illustrates how the recommendation system works with inputs and finally provides recommendation to end users.

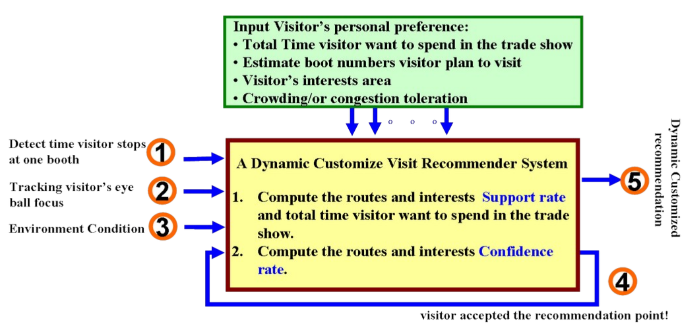

Figure 2 Input and output of the recommendation system

As we can see from Figure 2, the top block is the input of the visitor's personal preference which is used as criteria to our recommendation system. The input is predefined and provided by end users. The personal preference includes:

- Total time the visitor wants to spend in the trade show/exhibition/museum.
- Total number of booths the visitor plans to visit
- Visitor's interested area
- Congestion ratio tolerance.

Another input is from the real-time monitoring system which detects the following statistics:
- Time duration a visitor stops at one booth
- Visitor's eye ball focus
- Environment conditions

Utilizing inputs from personal preference and the real-time statistics, the recommendation system provides following outputs:

- The dynamically customized recommendations to the visitor by finding similar behavior/characteristic to the visitor
- Notify booth's owner when a candidate visitor is approaching or standing in front of the booth.

The recommendation system is trained by the decisions made by the user. If the user goes to the recommended booth, the data is sent back to the system and the system increase the weight of the rule (shown in Figure 2Error: Reference source not found direction 4).

PLANNING OF EXHIBITION VISITING JOURNEY

In this section, we use exhibition as example to demonstrate the recommendation system introduced in Section III, and how people can benefit from the proposed recommendation system.

Exhibition Scenario and Notations

- In this exhibition, there are five booths named A, B, C, D, E
- The time duration (interest level) that visitor spends in a booth is classified into three levels:
    - *l*(ow): spent less than 1 minute
    - *m*(edian): spent 1 minutes to 5 minutes
    - *h*(igh): spent more than 5 minutes
- Therefore, the following notation means
    - A_m: visitor stops at booth A for 1 to 5 minutes.
    - B_l: visitor stops at booth B less than 1 minute.
    - C_h: visitor stops at booth C more than 5 minutes.
- T(x) is the total time that visitor, x, plans to spend in the exhibition.
- Supporting rate is the percentage that the same visited booth path in the repository. The more people have visited the path, the higher supporting rate this path has.
- Confidence rate is a conditional probability, which is based on user's current visited history record and come up with how high the confidence of the record has.

Assume the system is running for a while and collecting records as follows:

User 1: [A_l, B_l, C_l, D_l, E_h]
User 2: [A_h, B_h, C_h, D_h, E_h]
User 3: [A_h, B_l, C_h, D_h, E_h]
User 4: [A_h, B_l, C_l, D_h, E_h]

All records are saved as set, describing that user visited which booth and the corresponding stop time spent (interest level) there.
The average time value of different interest levels is as following and it applies to all booths data:

$t(A_l)$ = 0.5 minute
$t(A_m)$ = 2.5 minutes
$t(A_h)$ = 7.5 minutes

User x visits the exhibition and would like to have recommended visiting plan provided by our recommendation system. User x already set his personal preference and his visiting history is tracked too.

[User x personal profile]
visit path and interest level : $[A_h, B_l]$
time plans to spend T(x) : 25 minutes
interest area : booth E

Supporting Rate of History Records and Filter Mechanism

First of all, the recommendation system engine calculates the supporting rate of each history record, shown in Table 1.

Table 1 Entries in the repository of the recommendation system

1	$[A_l, B_l, C_l, D_l, E_h]$	9.5	25%
2	$[A_h, B_h, C_h, D_h, E_h]$	37.5	25%
3	$[A_h, B_l, C_h, D_h, E_h]$	30.5	25%
4	$[A_h, B_l, C_l, D_h, E_h]$	23.5	25%
5	$[A_h, C_h, D_h, E_h]$	30	50%
6	$[A_h, B_h, C_h, D_h]$	30	25%
7	$[A_h, B_l, D_h, E_h]$	23	25%
8	$[A_h, B_l, D_h, E_h]$	23	50%
9	$[A_h, D_h, E_h]$	22.5	75%
10	$[A_h, C_h, E_h]$	22.5	50%

Secondly, the recommendation system engine performs filtering steps based on user x's preference and visiting record.

1. User x already visited booth A and B and corresponding interest level is $[A_h, B_l]$. Applying finding similar behavior/characteristic algorithm to the visitor, records 1, 2, 5, 6, 9, 10 are eliminated by the recommendation list.
2. User x plans to spend 25 minutes in the exhibition. Engine eliminated the records with time longer than 25 minutes. Record 3 is deleted.
3. System knows that user x is interested in booth E. The recommendation system will take booth E as criteria. Therefore, removing record 7 which does not include booth E.
4. As a result, the recommendation system has record 4 and 8 in the list.

Confidence Rate

To provide the top recommended visiting path to user x, the recommendation system applies 'confidence rate' algorithm to record 4 and record 8 listed in Table 1. The confidence rate result is shown in Table 2.

Table 2 Records after applying confidence rate algorithm

	History record (booth and corresponding level)	Total time spent	Supporting rate	Confidence rate
1				
2				
3				
4	$[A_h, B_l, C_l, D_h, E_h]$	23.5	25%	$P([C_l, D_h, E_h] \mid [A_h, B_l]) = 50\%$
5				
6				
7				
8	$[A_h, B_l, D_h, E_h]$	23	50%	$P([D_h, E_h] \mid [A_h, B_l]) = 100\%$
9				
10				

Finally, recommendation system engine sorts records from Table 2 based on confidence rate, and then total time spent and finally by supporting rate. Based on the sorting results, the system will provide record 8 [Ah, Bl, Dh, Eh] to user x. The recommended route is the most valuable one to user x because it highly matches user's preference and crowd behavior.

Congestion Condition

There will be 'rush hours' during a trade show. During rush hour, some booths may be crowded by people, which could cause increasing time visitors spent in this booth (for example, the visitor is waiting for available salesman or the booth is too crowded for the visitor to get out that area). So we introduce congestion ratio to illustrate the congestion condition.

CR(X) Congestion ratio of booth X =

the number of visitors in a booth X / booth area of X

CR(X) = 1 means people feel comfortable around a booth (normal time)

CR(X) > 1 means people feel 'crowded' around a booth (rush hour).

Due to congestion variation during a trade show, the time visitor spends in one booth does not actually reflect visitor's interest level in this booth. So we need to normalize the time using congestion ratio.

$t'(A_{userx}) = t(A_{userx}) * (1/CR(A))$, *where*

$t(A_{userx})$ *is the actual time user x spent in booth A and*

CR(A) is congestion ratio of booth A

For example, if user x spent 10 minutes in booth A and the congestion ratio in booth A is 2. It implies that user x takes more time staying in booth A than normal time because too many people are waiting for available salesman or user x is blocked in the booth. We need to normalize visitor time spent to reflect the 'actual' interest level of user x to booth A. Applying normalization formula:

$t'(A_{userx}) = 10 * (1/2) = 5$ *minutes.*

User x may spend only 5 minutes in booth A if booth A is not so crowded. We use the normalized value inserted into system repository. The supporting rate and confidence rate algorithm uses the normalized data to come up a more accurate recommendation for user.

Hot Spot Stack

If user prefers not to go to crowded booths, we can put the recommended booth into hot spot stack, and periodically check the booth congestion ratio. If the booth congestion ratio is less than a predefined value, system will notify user he/she may go to that booth he/she is interested in.

HotSpotStack(userx).push(D)

if CR(D) is larger than a threshold (ex. 2).

HotSpotStack(userx).pop(D) and notify user x

if CR(D) is less than a value (ex. 1.2),

Take the example provided before, the final recommended path to the visitor is [Ah, Bl, Dh, Eh]. However, at that time, congestion ratio of D R(D) is 6, which means booth D is crowded. Based on visitor's preference, if the visitor prefers not going to crowded booth, our system will put booth D to visitor's stack. When R(D) is less than a value (ex. R(D)<1.2), pop up a message and notify the visitor that he/she is okay to go to booth D now.

Time Effective Weight

The data of the last two hours is much more effective and valuable than data of 1-day or 2-days ago. It's more real data which reflects current trade show situation. Our system can use different weight based on records' timestamp.

Time effective weight W(record#):

	1	*if record # is taken less 2 hours*
W(record #)	*0.5*	*if record # is taken 2 hours ~ 1 days*
	0.1	*if record # is taken more than 1 days*

Multiplying the Time effective weight to rule's confidence rate, our system can provide more realistic recommendation to user.

The overall flow of the recommendation system, starting from collecting records, calculating supporting rate and confidence rate, filtering and sorting based on user's preference and other environment conditions, and finally providing a dynamic and highly customized visiting journey recommendation to user. Figure 3 is the flow chart of the recommendation system.

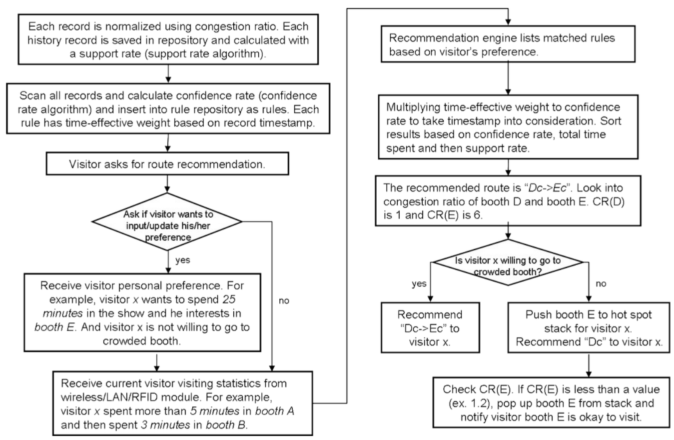

Figure 3 Overall flow of the recommendation system

CONCLUSIONS

This paper presents a location recommendation system based on historical behaviors of other visitors, leveraging the "Wisdom of Crowds". The recommendation system actively takes visitors' interests and preferences, real-time situations, such as geography, time, and other visitors' behaviors into consideration to provide accurate recommendations to visitors. It brings online recommendation experiences into real life and provide dynamic and customized recommendations. This system improves visitors' experiences and satisfaction during their visits to exhibitions or museums.

REFERENCES

Asthana, A., Cravatts, M., & Krzyzanowski, P. (1994, December). An indoor wireless system for personalized shopping assistance. In Mobile Computing Systems and Applications, 1994. Proceedings., Workshop on (pp. 69-74). IEEE.

Barnes Jr, M. L. (2013). U.S. Patent No. 8,611,919. Washington, DC: U.S. Patent and Trademark Office.

Barnes, M. (2002). U.S. Patent Application

Balabanoviæ, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. Communications of the ACM, 40(3), 66-72.

Bao, J., Zheng, Y., & Mokbel, M. F. (2012, November). Location-based and preference-aware recommendation using sparse geo-social networking data. InProceedings of the 20th International Conference on Advances in Geographic Information Systems (pp. 199-208). ACM.

Doughty, D., Ramer, J., & Soroca, A. (2007). U.S. Patent Application 11/929,016.

Gopalakrishnan, P., Roberts, R., Rathi, S., & Lortz, V. (2011). U.S. Patent Application 13/976,026.

Kuo, M. H., Chen, L. C., & Liang, C. W. (2009). Building and evaluating a location-based service recommendation system with a preference adjustment mechanism. Expert Systems with Applications, 36(2), 3543-3554.

Moore, J., Weiss, N., Lee, B. N., Sklar, M. E., & Shaw, B. (2012). U.S. Patent Application 13/415,277.

Oracle White Paper (2011, March). Next-Generation Product Recommendations.

Olugbara, O. O., Ojo, S. O., & Mphahlele, M. I. (2010). Exploiting image content in location-based shopping recommender systems for mobile users.International Journal of Information Technology & Decision Making, 9(05), 759-778.

Su, H. (2013). CrowdPlanner: A Crowd-Based Route Recommendation System.arXiv preprint arXiv:1309.2687.

Ye, M., Shou, D., Lee, W. C., Yin, P., & Janowicz, K. (2011, August). On the semantic annotation of places in location-based social networks. InProceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 520-528). ACM.