An Approach for Integrating Multidimensional Database into Context-Aware Recommender System

Mai Nhat Vinh, Nguyen Nhat Duy, Ho Thi Hoang Vy, and Le Nguyen Hoai Nam

University of Science, Vietnam National University - Ho Chi Minh City
{1012531,1012561}@student.hcmus.edu.vn,
{hthvy,lnhnam}@fit.hcmus.edu.vn

Abstract. Recommender system (RS) suggests useful items to users. Most of existing techniques in RS focus on using a rating matrix of users and items without considering recommendation context. In this paper, we take advantage of multidimensional database to present context information in context-aware RS. Therefore, we exploit the ability of OLAP aggregate operations to estimate user ratings. We propose an approach to translate the concepts of content-based, collaborative filtering and context recommendation into OLAP aggregate operations and integrate them to rating estimation function. Furthermore, through OLAP aggregate operations, our rating estimation function tends to solve the cold-start problem in RS and the data sparsity problem in context-aware RS. We develop a context-aware tour RS with our approach. In this system, we survey related researches to identify context information and user, item information being suitable to tour RS. We evaluate the system by accuracy and performance.

Keywords: context-aware recommender system, rating estimation, multidimensional database, multidimensional data cube, OLAP operation.

1 Introduction

When we deal with huge amounts of information, we are confused about making decisions. A recommender System (RS) plays a role as an expert supporting us to solve the problem of overwhelming information easily and quickly. The recommender system has attracted much attention in information science because there are still many open issues yet to be resolved. Most of the researches on the recommender system often focus on algorithmic improvement or propose new techniques to suggest useful items to users. Recommender systems are software agents that elicit the interests and preferences of individual consumers and make recommendations accordingly [12]. They have the potential to support and improve the quality of the decisions of users while searching for and selecting items.

Three techniques for suggesting useful items in RS are collaborative filtering, content-based and hybrid [3][7]. They are mainly based on the user habits in the past or based on who have the same interests with them in the past to predict the preference that user would give to an item.

K. Saeed and V. Snášel (Eds.): CISIM 2014, LNCS 8838, pp. 231–242, 2014.

The collaborative filtering technique uses the experience of a group of users to recommend items. It based on the assumption that users will give ratings to items implicitly or explicitly and users who have similar preferences in the past will have similar preferences in the future. The input of the system is only a matrix of user ratings on items. The output could be a (numerical) prediction indicating to what degree the current user will like or dislike a certain item or a top-N list of recommendation items. However, the weakness of this technique is that the system cannot provide a good recommendation if data is sparse. It means that the number of available ratings could be too small to compute a recommendation. The main reason for that problem is that users do not always provide ratings to items. Another problem is when new item or new user has just been entered to the system, the system has no its historical information to generate recommendation.

For content-based technique, the recommendation is based on the consideration of characteristics of the item that a user has preferred in the past. Consequently, users will not be able to get novel items which are completely different from the available items in the system. Especially, when a user has changed the current preferences, the system cannot give this user the best recommendation. Furthermore, a new user who has just been entered to the system has no its historical information for the system to generate recommendation.

Most of existing approaches have focused on problems suggesting related items for users without considering recommendation context. There are many researches on the CONTEXT definitions and one of general definitions is "conditions or circumstances which affect something" [1][4]. In personalized Recommender Systems, context information has been recognized as an important factor to be considered [1][2][3][4][5][6]. Therefore, Adomavicius et al. in [1] propose a multidimensional approach for presenting context information beside the information of users and items in RS in order to increase the quality and performance of recommendation. Concretely, this approach presents user, item and context information as dimension tables in data warehouse. Information about user ratings to an item in a specific context is stored in a fact table. Hence, when multidimensional database is built into multidimensional data cube from dimension tables and fact table in data warehouse, we take advantage of the benefits of OLAP operations on data cube with aggregate ability to compute user rating for an item on a specific context. The user rating computation for an item on a specific context in this approach presented by Adomavicius et al. in [4] is based on the content-based technique. For example, we may want to know how individual user "Mr.A" like a destination place "Nha Trang City", we need to compute R(Mr.A, Nha Trang) that is the predicted rating of "Mr.A" to "Nha Trang". Because beach is a feature of the "Nha Trang" destination, we can use the hierarchy that "beach"is higher level in destination place dimension to compute aggregate rating R(Mr.A, Beach). Concretely, R(Mr.A, Nha Trang) is computed by OLAP rollup operation on destination place dimension as following:

$$R(\text{Mr.A},\text{NhaTrang})=R(\text{Mr.A},\text{rollup NhaTrang})=AGGR_{x.\text{feature}=\text{Beach}}R(\text{Mr.A},x)= R(\text{Mr.A, Beach})$$

Inferentially, R(Mr.A, Beach) is considered as R(Mr.A, Nha Trang).

In [6], the authors propose a rating estimation function for user u to item i in a certain context c referred as r(u, i, c). That function in [6] is based on: (1) historical ratings of user u to all items being similar to item i referred as aggregate computation R(u, rollup i, c) like content-based concept or (2) historical rating of all users being similar to user u to item i referred as aggregate computation R(rollup u, i, c) like collaborative filtering concept.

However, we realize that:

- When a new user u_{new} is newly added to the system, if only using (1), the estimative rating of u_{new} for any items will have the result being equal to 0 because there is no information of purchase history of u_{new}. Or if only using (2) to estimate any user's rating for new item i_{new} , which is newly added to the system, the rating estimation function will also have the result being equal to 0 because there is no purchase history of any users for i_{new}. Therefore, we propose integrating (1), (2) in this approach to solve these problems. However, this integration is not enough when a new user and new item is added into the system, the recommender system could not estimate the rating of u_{new} for $i_{new.}$ Hence, we propose component (3) to help to solve this problem, (3) is historical ratings of all users being similar to user u to all items being similar to item i referred as aggregate computation R(rollup u, rollup i, c). This integration should be translated into OLAP aggregate operations on multidimensional data cube of a multidimensional database.
- Furthermore, with the approach for presenting user, item and context information as dimension tables, how to design every dimensions including selecting dimension attributes to hierarchize a dimension.
- Applying context information to compute rating will increase the accuracy of recommendation but it also narrows down the set of data used in computing recommendation. This leads the case that there are few ratings for prediction. That is data sparsity problem in context-aware RS. Especially, if there are no ratings in context c, r(u, i, c) will be equal to 0.

In this paper, we propose an approach to solve three above problems when integrating multidimensional database to context-aware RS. We verify this solution by developing a context-aware tour recommender system.

2 The Approach

In our approach, user, item and contextual information are presented by a multidimensional data model with snowflake schema [10] as Figure 1. The aim is to take advantage of the aggregation and hierarchy with OLAP operation on multidimensional data cube in this model.

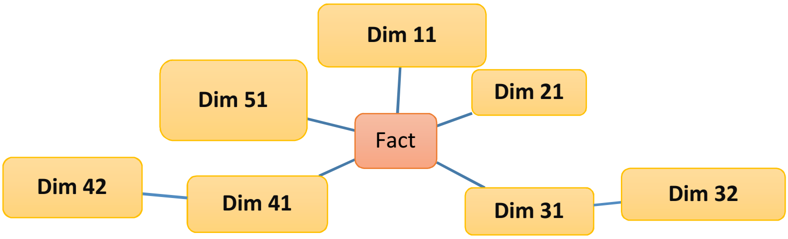

Fig. 1. Snowflake schema in multidimensional data model

In Figure 1**,** the fact table contains a measure presenting user rating with related dimensions. Each entity including context, user and item information will be presented by one dimension.

With the above model, we suggest a rating estimation function under OLAP aggregate operation on data cube of multidimensional database. Concretely, our rating estimation function for user u to item i in a specific context c referred as r(u,i,c) is as following:

$$r(u, i, c) = w_1 \times R(\text{rollup } u, i, c) + w_2 \times R(u, \text{rollup } i, c) \qquad \text{(F1)}$$

With r is prediction rating, R is available ratings in the system by user-supplied. Total of weights w_1 and w_2 must equal to 1.

As mentioned above, our rating estimation function will integrate (1) and (2). This integration is not only for solving the new user and new item problems but also for gaining better performance with fewer of the drawbacks of any individual one [18].Some of the combination methods that have been used such as Weighted, Switching, Mixed, Feature combination, Meta-level. Concretely, in this paper, we select the weighted method, which is shown in F1.Using weighted method, in our rating estimation function, the combination is linear, each element is assigned a weight and the weights can be adjustable according to user's feedbacks. Furthermore, these weights indicate the degree of each element in computing recommendation. Typically, if the weight of element R(rollup u, i, c) is higher than the weights of the others, that means that user u tends to get advices of the users who similar to user u. If the weight of element R(u, rollup i, c) is higher than the weights of the others, that means that user u tends to the items which user u liked in the past. For example, when a user wants to travel somewhere, he/she may consider asking guides from friends, relatives with basing on the features similar to destination places that he/she travelled before.

If our rating estimation function just integrates (1), (2), when a new user u_{new} and new item i_{new} is added into the system, the recommender system will not be able to estimate the rating of u_{new} for i_{new}, so r(u_{new}, i_{new}, c) is equal to 0. Because, when i_{new} is just added into the system, there is no rating of any users for i_{new}, so function R(rollup u_{new}, i_{new}, c) will be equal to 0. Similarly, when u_{new} is just added into the system, there is no rating of u_{new}, so function R(u_{new}, rollup i_{new}, c) will be equal to 0. Hence, we propose component (3) will help solve this problem. (3) is historical ratings of all users being similar to user u to all items being similar to item i referred as aggregate computation R(rollup u, rollup i, c). In this case, our rating estimation function is as following:

$$r(u, i, c) = R(\text{rollup } u, \text{rollup } i, c) \quad \text{(F2)}$$

Another important issue is how to design every dimensions including selecting dimension attributes to hierarchize each dimension. This depends on the considered dimensions in a certain domain. For example, in the tour recommender system, user dimension should be hierarchized by age or personality attribute because age or personality of users influence their selection of travel destination. Furthermore, the users who have the same age or the same personality often tend to like the same category of destination places. Therefore, user dimension may have two sub-dimensions as in Figure 2:

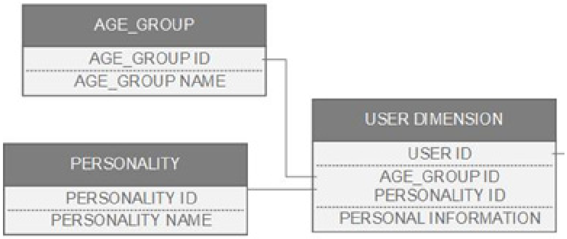

Fig. 2. Example of dimension design

When we roll up on the user dimension R(rollup u, i, c), we should consider rolling up on age or rolling up on personality or both. Our solution is to compute the average of all of them.

As mentioned above, considering user rating to items in a specific context may lead that prediction rating may be equal 0 if in the context which there is no rating or only a few rating. That is sparsity problem in context-aware RS. Hence, we propose to solve this problem by estimating rating in a context being similar to current context in this case. The similar context is the higher level of current context c and computed by rollup operation in context hierarchy. In this case, rating estimation function r(u,i,c) is described as following:

$$r'(u, i, c) = r(u, i, \text{rollup } c) \quad \text{(F3)}$$

3 Our Context-Aware Tour Recommender System

3.1 System Model

The proposed approach is applied to build a context-aware tour RS. The architecture of our tour RS is illustrated in Figure 3. It includes three modules:

- In the first module, the system will automatically get the tours from tourism websites then integrate and transfer them to data warehouse. This module is operated many times at regular intervals.

- The second module is the recommendation engine. Its input is context information from user. It is responsible for computing prediction rating by using the rating estimation function to generate a list of destination places that is the most suitable with the user. Then, the engine will search a list of tours in which the number of suitable destination places are maximal.
- The third module displays the recommendation to the user and receives user feedback.

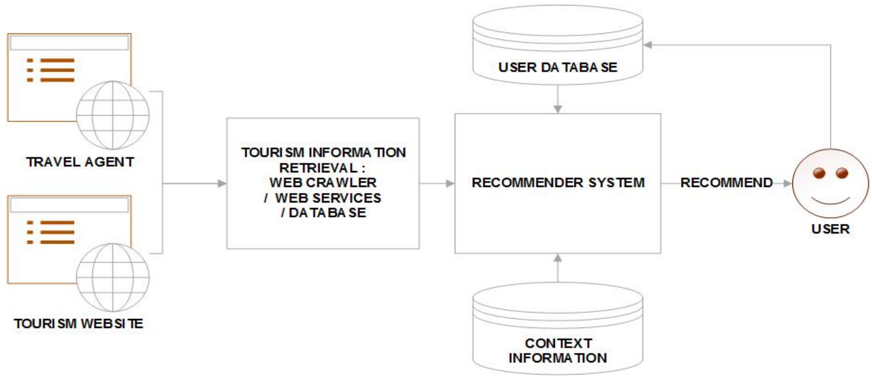

Fig. 3. Our tour recommender system model

3.2 Tour Context, User, Item Information

Based on [6][8][11][13][14][15], we summarize the criteria effecting to make decision of users who want to select some destination places to travel, then classify them into three groups: User, Item and Context to apply them into our context-aware tour RS. In Table 1, we illustrate some criteria that can be considered as context in tour context-aware RS.

Table 1. Criteria consideredas context

Context	Description	References
Time	**Season** :When should you travel to Japan? Eg: You should travel to Japan in Spring season because it got cherry-blossom in this season. **Festival** : When should you travel to Brazil ? Eg : You should travel to Brazil during Carnival time .	[15]
Goal	Which destinations should you travel for discovering Korea culture? Eg: You should find an area being well-known with the Kim Chi specialty	[8], [15]
Companion	If you travel with your lover where you should travel to? Eg: You and your lover may like a romantic scene at Themes river, London	[13], [15]

Table 1. *(Continued)*

Transport	By which transport you should travel to Pattaya island in Thailand? Eg: You ought to travel by boat to contemplate the coral	[15]
Weather	Which is the best weather to travel Honolulu? Eg: You should travel in summer time or when it's sunny	[15]
Distance	You are in USA. You want to travel to the beach. There are many choices for you such as Lanikai, Hawaii, Turquoise Bay, or Australia. They're all both exciting but you have to consider because it's a so far journey from USA to Australia.	[15]

In Table 2 and Table 3, we summarize criteria being attributes of user dimension and Destination dimension.

Table 2. Criteria considered as attributes for User dimension

Attribute	**References**	**Is hierarchy attribute on the dimension?**
Favorite	[6]	Yes
Age	[6], [8], [11], [13], [14]	Yes. This attribute highly effect to user 's decision mentioned in Table 1 in [8]
Sex	[6][8]	
Income	[8]	
Education level	[8]	

Table 3. Criteria considered as attributes for Destination dimension

Attribute	**References**	**Is hierarchy attribute on the dimension?**
Destination Category	[6], [13], [15], [16]	Yes
Region	[15]	Yes
Activity	[6], [13], [15], [16]	

3.3 Data Model

We had made the summary of important attributes that highly effects user decision. We identify a specific list of context dimensions and user, item dimension to apply to our context-aware tour RS as in Figure 4. Time context dimension should be separated into two dimensions which are season and festival as said in table 1. We differentiate festival from season dimension because we analyzed that the weather of the places depends on season so it will strong affect to user travel preferences and the festival dimension is either. In big festival occasions such as in Viet Nam, at Tet holiday, woman's day, Da Lat flower festival, etc…there are many special events than other occasion and users may be attracted. Therefore, in our system, we decide to include three contexts that are season (eg: summer, spring…), festival (eg: Tet holiday…) and companion because we couldn't collect enough data to the others context dimension.

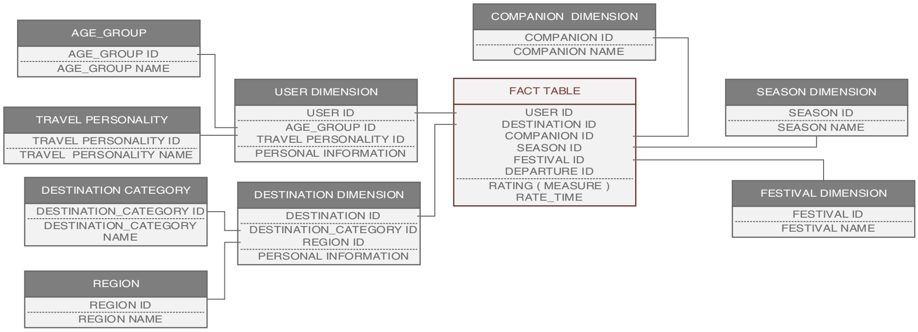

Fig. 4. Data model to build multidimensional cube of ratings

About rating data, we surveyed a number of participants by an online survey. At last, there are 5600 ratings were entered by 92 participants in two months from April 2014 to June 2014.The content of this survey includes two part: the first part with 8 questions about age and travel personality; the second part with 100 travel destinations belonging to 30 categories in 8 region, participants were required to provide the rating and answer the question related to context information as following: when and who you go with?. Overall, 89 participants responded to us and 5600 rating were collected. The collected data were transferred to multidimensional cube and ready for the estimation process.

4 Evaluation

4.1 Execution Time

We do experiment to measure the execution time of the system for estimating user rating. The execution time is taken in two data model of multidimensional database. For the first model, multidimensional database is presented by a relational data model in which context, user, item, rating are tables with foreign keys. For second model, multidimensional database is a multidimensional data cube which is built from relational data model of the first model.

The experiment is on HP 430 Core i5-2450M Sandy Bridge (2.5GHz up to 3.1GHz, 4GB 1333 MHz DDR3 SDRAM , 500GB), SQL server 2012. In relational data model, we use *System.Data.SqlClient* in .Net to execute queries to database engine in SQL server while in multidimensional data cube, *Microsoft.AnalysisServices.AdomdClient* is used for executing OLAP operation to data cube.

The result of the experiment is shown in Figure 5, when the number of rating records is less than 20000, the execution time in the relational data model is better than the execution time of the multidimensional data cube. However, when the number of rating records is more than 20000, the execution time of the multidimensional cube is far better than the execution time of the relational data model. Furthermore, when the number of rating records increases, the execution time in the relational data model increases quickly and reaches 109.932 milliseconds at 1 million rating records while the increase of the execution time of the multidimensional data cube is not considerable. The execution time of the multidimensional data cube is 7.414 milliseconds at 1 million rating records. The main reason is that when multidimensional database is presented by relational data model, rollup operation is executed by join operation on related tables, so the more the number of rating records, the more the execution time of rollup operation on it while multidimensional data cube specializes in supporting executing rollup operations

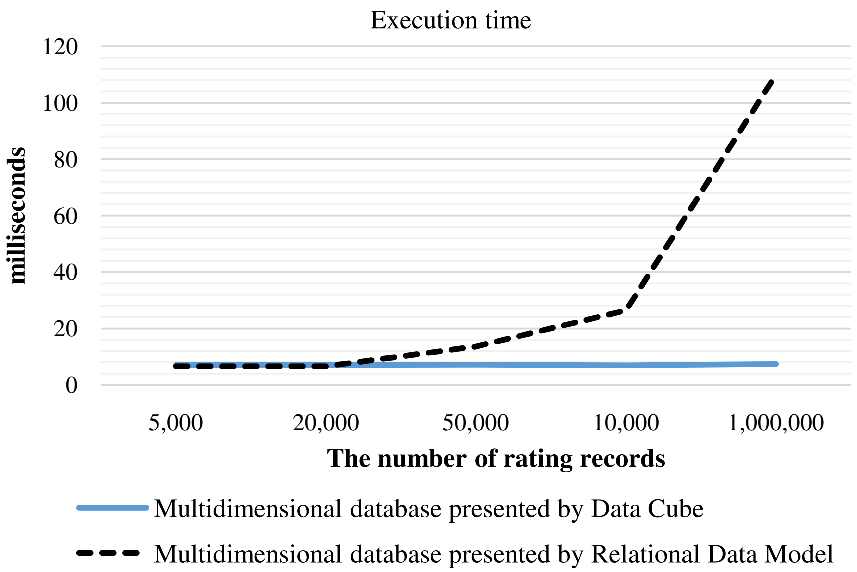

Fig. 5. The execution time of the system for estimating user rating

4.2 Accuracy

Using the metric in [17], the system accuracy was computed by counting the number of useful recommendations. Useful recommendations are collected by getting user feedback after receiving the recommendation. Concretely, the participants are required to enter context information then the system will predict a list of top 10 destination places being suitable to them. They then answer the number of recommended destination places satisfying their needs. There are 89 participants joined in the evaluation but till now, 21 responses out of 89 were collected. Currently, the system accuracy is 80.4762 %.

We also have used Mean Absolute Error (MAE) to compute the deviation between predicted ratings and actual ratings. Currently, just a few responses were collected from participants. However, we will continue doing the evaluation with more participants.

5 Conclusion

We desire to highlight the importance of context information in RS. Multidimensional database is one of suitable solution for presenting context information. Furthermore, multidimensional data cube model of multidimensional database facilitates OLAP aggregate operations. Therefore, we translated rating estimation function into OLAP aggregate operations on multidimensional data cube. Under OLAP aggregate operations, rating estimation function is the integration and overcoming of the existing technique in RS. About the data sparsity problem that means there are few user ratings to estimate, our solution is to extend the computing recommendation space to historical rating of all users being similar to current user to all item being similar to current item.

We developed a context aware tour RS with the proposed approach. We surveyed related researches to identify context, user, item dimensions and then design dimension hierarchies. Because there are not standard dataset for our context aware tour RS, we collect user ratings from a number of users for preparing rating estimation.

We do experiment to measure the execution time of the system in rating estimation. Moreover, we compare the execution time in two data model of multidimensional database which are relational data model, multidimensional data cube model. The comparison shows that when the number of rating records increases, the execution time with relational database model increases quickly and reach 109.932 milliseconds at 1 million rating records. With multidimensional data cube model, the increase of the execution time is not considerable when the number of rating records increases. The execution time with multidimensional data cube model is 7.414 milliseconds at 1million rating records. Hence, we conclude that multidimensional data cube model is better than relational data model in case of presenting multidimensional database in context-aware RS.

The system accuracy is 80.4762 %. Currently, we have been doing evaluation on higher number of users.

6 Future Work

Next time, we will focus on optimizing system performance by studying issues related to multidimensional data cube. Furthermore, we are trying to combine other methods to make clear the user preferences. Clearly, the more recommender systems understand user preferences, the better recommendations they can provide. Recommendation must base on user preferences. Therefore, understanding user preferences is very important but it's a really hard work. Modeling of user preferences needs their relevance feedback on the recommendations [9]. The relevance feedback may be collected either explicitly or implicitly. The explicit feedback indicates the rating user provided. However, it wastes user‘s effort, time, and cost because users have to stop their action to enter explicit rating, users don’t like to give rating. So we can observer user behaviors such as time spent on viewing the items, number of accesses to an item, or the action click to view, like, shared, buy items… The implicit feedback can reduce the cost of rating items by saving the user’s time, however there remains a computational cost in storing and processing the implicit rating data, this can be hidden from the user. Therefore, we can improve system performance and overcome the sparsity problem by combine them together. Moreover, we also integrate other techniques such as indexing, partition data on multidimensional database to improve the performance.

Acknowledgments. This research is supported by research funding from Advanced Program in Computer Science, University of Science, Vietnam National University - Ho Chi Minh City.

References

1. Adomavicius, G., Tuzhilin, A.: Multidimensional recommender systems: A data warehousing approach. In: Fiege, L., Mühl, G., Wilhelm, U.G. (eds.) WELCOM 2001. LNCS, vol. 2232, pp. 180–192. Springer, Heidelberg (2001)
2. Araque, F., Salguero, A., Abad, M.M.: Application of data warehouse and Decision Support System in soaring site recommendation. In: Information and Communication Technologies in Tourism, pp. 308–319 (2006)
3. Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering 17(6) (2005)
4. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach. ACM Transactions on Information Systems (TOIS) 23(1), 103–145 (2005)
5. Adomavicius, G., Tuzhilin, A.: Context-Aware Recommender Systems. In: Recommender Systems Handbook. Springer (2011)
6. Rahman, M.M.: Contextual Recommender Systems Using a Multidimensional Approach. International Journal of Emerging Technology and Advanced Engineering 3(8) (August 2013)
7. Candillier, L., Jack, K., Fessant, F., Meyer, F.: State-of-the-Art Recommender Systems. In: Collaborative and Social Information Retrieval and Access: Techniques for Improved User Modeling, pp. 1–22. IGI global (2008)
8. Fang-Ming, H., Yu-Tzeng, L., Tu-Kuang, H.: Design and implementation of an intelligent recommendation system for tourist attractions: The integration of EBM model, Bayesian network and Google Maps. In: Fang-Ming, H., Yu-Tzeng, L., Tu-Kuang, H. (eds.) Expert Systems with Applications, vol. 39. Elsevier (2012)
9. Seo, Y.-W., Zhang, B.-T.: Learning User’s Preferences by Analyzing Web-Browsing Behaviors. In: Proceedings of the 4th International Conference on Autonomous Agents, Barcelona, Spain, pp. 381–387 (2000)
10. Inmon, W.H.: Building the Data Warehouse. John Wiley (2002)

11. Schiaffino, S., Amandi, A.: Building an expert travel agent as a software agent. In: Expert Systems with Applications. Elsevier (2009)
12. Xiao, B.: E-commerce item recommendation agents: Use, characteristics, and impact. MIS Quarterly 31(1), 137–209 (2007)
13. Ananthapadmanaban, K.R., Srivatsa, S.K.: Personalization of user Profile: Creating user Profile Ontology for Tamilnadu Tourism. International Journal of Computer Applications (0975 – 8887) 23(8) (June 2011)
14. Hanlan, J., Fuller, D., Wilde, S.J.: The travel destination decision process and the relevance of segmentation studies to the marketing of regional tourism destinations in an Australian context. In: Center for Enterprise Development and Research Occasional P aper, no. 1, Centre for Regional Tourism Research, Southern Cross University, Coffs Harbour, NSW (2005)
15. Kabassi, K.: Personalizing recommendations for tourists. Telematics and Informatics 27, 51–66 (2010)
16. Agarwal, J., Sharma, N., Kumar, P., Parshav, V., Srivastava, A., Goudar, R.H.: Intelligent Search in E- Tourism ServicesUsing Recommendation System: Perfect Guidefor Tourist. In: 2013 7th International Conference on Intelligent Systems and Control (ISCO), India (2013)
17. Olmo, F.H., Gaudioso, E.: Evaluation of recommender systems: A new approach. Journal Expert Systems with Applications: An International 35(3) (October 2008)
18. Burke, R.: Hybrid Recommender Systems: Survey and Experiments (2009)