International Journal of Computer and Information Technology (ISSN: 2279 – 0764)
Volume 06 – Issue 05, September 2017

Cold Start Problem Solving Exploiting Social Network Information

Anongporn Salaiwarakul, Wanarat Juraphanthong, Kraisak Kesorn*

Computer Science and Information Technology Department, Science Faculty,
Naresuan University, Phitsanulok, 65000, Thailand

**Email: kraisak [AT] nu.ac.th*

***Abstract*—Cold start problem is a well-known problem for recommendation system (RS) when the initial data is not adequate for user preference analysis which usually happen with a new user, so called cold start user problem. Typically, the system has to ask users to input some initial data e.g. personal data to allow the RS to start suggestion some products to a user. However, this method causes additional task to a user and is usually ignored because of security reason. This paper presents a novel technique to exploit social network data to solve such a problem. The sophisticated framework is developed to extract user data available from Facebook and analyze user preference in the domain of Tourism and then recommend some interesting attraction to a user. The experimental results illustrates that Check-in data from Facebook is very useful for cold start problem solving and then effectively recommendation performance can be achieved.**

***Index Terms*— Social network; Cold-start problem; Recommendation system; Personalization**

I. Introduction

The main challenge of RSs is that user information is insufficient for user preference analysis [1], which is usually known as the “cold-start” problem. Two types of cold-start problems are addressed by Adomavicius *et al.* [2]: 1) cold-start items and 2) cold-start users. The first problem occurs when there are insufficient previously submitted ratings about products that will be suggested to users. The second problem happens to a new user and RSs cannot recommend products to new users because the absence of previous user data; therefore, it is not possible to analysis user preferences and unable to make robust recommendations [3]. As such, the RS needs to ask some personal information in order to analyze and recommend some related products that meet the individual user’s preferences. Huge information available in social network, e.g. Facebook, is one of the potential resources to overcome such a problem. There are many activities, e.g., comments, likes, and check-in which can represent a rich source of knowledge about user preferences. Thus, we can exploit this information to solve the cold-start user problem. As a result, social networking information is used to not only solve the explicit data-acquisition problem, but also the cold-start problem and, consequently, improve the prediction accuracy of the RS.

This paper introduces a framework, which has been developed to contribute to the cold start problem using information available on Facebook. The main novelty of this paper is that Facebook friends’ check-in data are aggregated and exploited for personalized attraction recommendations, which makes the system more robust against the cold-start user problem using only a single type of data. We hypothesized that information from alternative external sources can substitute or complement missing data to facilitate accurate recommendations [4] to overcome the cold-start problem. Therefore, the huge amount of personal data available from Facebook could be used as a valuable external source to solve this problem.

The remainder of this paper is organized as follows. Section II presents our proposed technique to overcome cold-start problem. Section III shows our experimental results and discussion. Finally, section IV concludes our key novelties, limitations, and further work.

II. Proposed framework

To overcome the cold start user problem, we present the PTIS (Personalized Tourism Information Service) framework that recommends attractions to tourists based on using Facebook check-in data. If personal data of a user is inadequate, check-in information of his friends will be used to find his interests.

A. Friend Interaction Computation

Friend interactions are analyzed using the friend analysis algorithm, ay-fb-friend-rank [5] to separate close friends (more interactions) from others which deploys an EdgeRank technique [6] to perform such a task. This technique comprises three components: affinity score, edge weight, and time decay.

1) The affinity score is the interaction score. For example, Jane often writes on John’s wall and thus Jane will have very a high affinity score with John. The affinity score is computed based on: (i) explicit actions that users do (e.g., clicking, liking, commenting, tagging, sharing, and mutual friends); (ii) the proximity of the person who took the action in relation to

user; and (iii) how old of the action they took.

2) Edge weight gives various weights to various actions. Edge is created from every action of a user and each of those edges, apart from clicks, creates a potential story. For example, comments have higher edge weight than likes.

3) Time decay adjusts the score of the story according to time. When a story gets older, its score is lower. Typically, newsfeed is usually populated with edges that have the highest score at that very moment in time when a user logs into Facebook. Based on the idea of those algorithms, friend interactions can be defined as:

Definition 1. Friend interaction (F) of user (u) that shows the level of interaction between a user and his or her friends. F is a set of friend-interaction score pairs:

$$F(u) = \{(f,s) \mid f \in \Re \text{ and } s \text{ is in the range } [0,1]\},$$

where f is a friend in Facebook ($\Re$) and s is the interaction score, which varies from 0 to 1. The friend-interaction level (F) is in the range of 0 and 1, where 0 means that users have no interaction between them whereas 1 shows that they have very high interaction.

B. User-interest Analysis

There are two situations for extracting Facebook check-in data to identify user preferences: adequate and inadequate information for PTIS. Adequate information indicates to users with check-in data greater than the threshold to analyze user interests and inadequate information refers to users with insufficient check-in data. In the latter case, the system needs to acquire data from friends of a user on Facebook. To identify user interest based on personal data from Facebook, the computation scheme of user interests can be defined as (1), where $I(c)$ is an interest level in category c, n_c is the check-in numbers for a category c and $I(c)$ is normalized to the range of 0 and 1.

$$I_u(c) = \frac{n_c}{\sum_{i=1}^{6} n_i}, \; 1 \le c \le 6, \qquad (1)$$

To analyze user interest based on information from Facebook friends, the computation scheme from (1) can be modified as shown in (2). The interest level extrapolation of user u can be calculated from an aggregation of Facebook friends' check-in data. F_i is the level of interaction between users and friends, ith, $I_i(c)$ is the interest level of each category of friends, ith, n refers to numbers of friends in Facebook, and $n \in N$ where N is a set of close Facebook friends. The $I_u(c)$ value is scaled to the range of 0 and 1.

$$I_u(c) = \frac{aggr\, F_i I_i(c)}{\sum_{i=1}^{n} F_i}, \; i \le n \qquad (2)$$

A user-interest analysis algorithm is shown in Algorithm I. Having obtained user interests, they are further used to construct a user model for attraction recommendations. There are several approaches to construct user model e.g. ontology [7], [8] and statistical model [9]. The user model is constructed to store user interests (e.g., personal data, interest in attractions, feedback information, and interactions between users and PTIS) in a structured model (RDB), called "user profile". This data is useful for future suggestions when the user returns. RDB is chosen because its consistency, integrity, easy maintenance (insert, update, or delete), and better security than flat file or ontology-based models.

Algorithm I. User interest analysis

```
Get all Facebook user check-ins.
Remove duplicated check-ins from the same day.
Remove any check-ins that are not attractions.
Classify check-ins into six categories.
    If all classified check-ins of a user ≥ threshold
        Compute level of interest $I(c)$ using (1)
    Else
        Identify close friends.
        Get all Facebook check-ins of close friends
        If classified check-ins of close friends ≥ threshold
            Compute level of interest $I(c)$ using (2)
        Else
            Define level of interest $I(c)$ with average value from popularity.
Sent $I(c)$ to next module.
```

C. Venue Recommendation

The presented method was applied to recommend attractions based on the analysis of user check-ins by matching the characteristics of the venues with the user characteristics [10]. This process consists of two main steps:

1) Attraction weight computation

This process computes the important of all attractions in each category for choosing the top $R(c)$ places to a user. The computation method has three parameters: (i) Place popularity (P): Almost all tourists typically want to visit the iconic places at the destinations. The popular places are acquired by extracting the number of check-ins from Facebook. The higher number of check-ins a destination has, the more popular it is. (ii) Visited places by friends (F): Tourists usually are also interested in places where their friends have been. Therefore, this parameter is also important and could significantly improve recommendation accuracy. Based on these parameters, the attraction weighting uses a linear regression analysis model for the attraction recommendations as shown in (3).

$$W(p)=\left[\alpha P(p)+\beta F(p)\right] \ , \qquad (3)$$

where $W(p)$ is the attraction weight, α is popularity score, β is the weight of Facebook check-ins at places by user friends, and γ is the score of the appropriate time for visiting attractions. Popularity of place, $P(p)$, is measured by the number of Facebook likes and check-ins for a place. $n_{ch,p}$ is the number of Facebook check-ins at a place and $n_{li,p}$ is the number of Facebook likes for a place. Max is the highest number of Facebook check-ins and likes for each attraction. We normalize the range of check-in and like variables to ensure that the data are not overloaded by each other in terms of distance measures [11] as shown in (4).

$$P(p)=\frac{n_{ch,p}}{\max(n_{ch,p})}+\frac{n_{li,p}}{\max(n_{li,p})} \ , \qquad (4)$$

Tourists are often interested in places where friends have visited. Therefore, we also consider this parameter. Places visited by friends, $F(p)$, is measured by the Facebook check-ins by close friends as shown in (5). F_i is the level of interaction between users and close friends i^th^, C_i =1 if friend i^th^ has checked-in on Facebook at this place p and C_i =0 otherwise.

$$F(p)=\frac{\sum_{i=1}^{n}F_iC_i}{\sum_{i=1}^{n}F_i} \ , \qquad (5)$$

2) Result ranking

This process calculates the score of attractions that relevance to user preference in the user model. The recommended attractions are ranked by descending order and deliver a list of the top-*N* ranked attractions that the user may prefer. The attraction ranking is measured by (6). If $Rank\ (p_1)$ has a greater score than $Rank\ (p_2)$, this indicates that p_1 has more relevance to user interest than p_2.

$$Rank(p)=\frac{R(c)\times W(p)}{N} \ , \qquad (6)$$

where N is the number of attraction recommendations.

III. Experimental results and discussion

Several experiments have been performed to evaluate the presented system using a dataset extracted from volunteers with active Facebook accounts. We recruited volunteers by invitation because we wanted to control for demographic features (age, gender, education, and marital status) and level of Facebook activity. However, the ages of participants in this experiment were not normally distributed. Participants came from Naresuan University (NU), Thailand and the majority was aged between 18–30 years, whereas only 10% of all participants were older than 35 years working in NU. This is because a very small number of elderly people have Facebook accounts and lack information technology experience. For the whole dataset, the number of users and attraction check-ins is 120 and 12,500 respectively. All experiments were undertaken in the same group of volunteers. The evaluations are compared to state-of-the art frameworks such as the content-based and collaborative-based approaches as well as theoretically compared the performance of the PTIS with other RS systems through the discussion of our experimental results. The standard measures, average precision, and rank score, are deployed to evaluate the recommendation efficiency of PTIS. We conducted two experiments: Experiments 1 determined the recommendation performance of the system using Facebook check-in data, and Experiment 2 evaluated the optimal numbers of friends when the system faces a cold-start problem.

A. Evaluation of Check-in Data

We hypothesized that users' Facebook check-in information can represent their preferences and overcome the cold-start problem. Therefore, this evaluation studies the effectiveness of using check-in data to tackle such the problem. To evaluate the hypothesis, three cases are investigated. The first case uses personal data of a user to analyze his or her interests, called *personal data case*. To study the cold-start problem, check-in data of some volunteers was deleted, which leads the system to use data from others by gathering friends' data from Facebook. This scenario is the second case in this experiment, the so-called *friend data case*. The final case uses the popularity information of attractions considering from the number of Facebook check-ins. A higher number of check-ins indicates the greater popularity of the attraction. This is called the *popularity case*. This information is aggregated from various types of data e.g. number of check-ins, likes, and shares of each attraction. Volunteers participated in the study by manually ranking the top-*10* attractions in each category that they were interested in, which is called *true ranking*. We compared the recommendation results generated by the system to those in the true ranking. As shown in Figure 1, the average ranking accuracy was up to 83.57% and 76.37% for the personal and friend cases, respectively. These results demonstrate that recommendation accuracy relying solely on Facebook friend data is slightly lower than the results obtained from using individual data. This confirms that personal data has higher quality than friends' data to represent user interests. However, both cases can significantly increase the precision of results compared with using popularity data, which achieved only 60.64% accuracy. Because of its high sparsity of personal data, the popularity case cannot effectively represent user interests as well as when using personal data. Figure 1 depicts the preference prediction performances of the system deployed using user individual data, friend data, and popularity data. Because extracting check-in data is expensive and time consuming, more experiments were conducted to examine the optimal number of check-ins for precision compared with time spent in data extraction.

As depicted in Figure 2, ranking accuracy tends to be higher when the numbers of check-in data and friend data increase. In the case of using user personal data, the average recommendation accuracy is up to 88.38% when the system exploits latest 40 or more check-in different attractions, whereas there is 78.49% of average prediction accuracy using friend data. Therefore, these trends demonstrate that the more check-in data are exploited, the higher the recommendation performance is obtained. However, this is not always the case because the number of check-ins and ranking performance is not always directly variant. Based on the results shown in Figure 2, ranking performance trends to slightly decrease when the number of check-ins is more than 40.

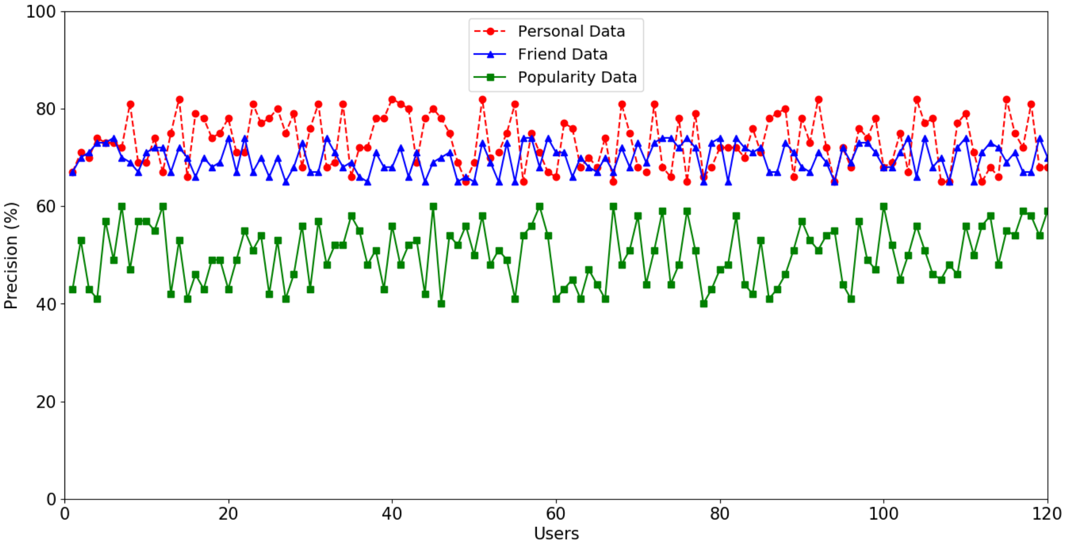


Figure 1: 120 users' interest prediction accuracy using three different data sources: personal, friend, and popularity data.

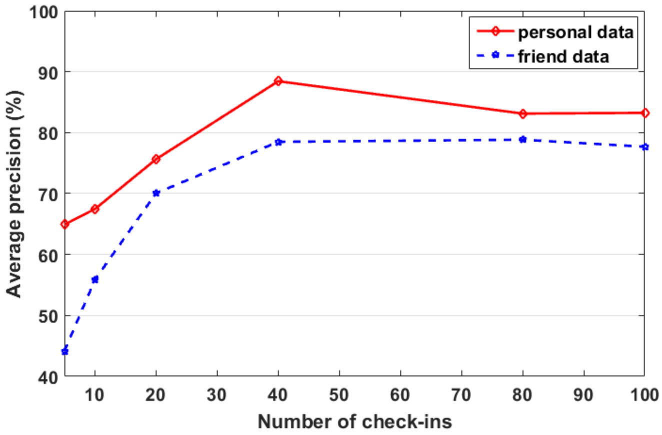


Figure 2: The category ranking accuracies with different numbers of check-ins using personal and friend data.

We analyzed this result and found that data from too many friends can introduce noise in the user-interest computation model, which can lower the accuracy of the PTIS. We also found that using all check-in data of a user can cause imprecise recommendation because they do not represent currently user interests. Therefore, latest 40 check-ins is the optimum value for the PTIS in the tourism domain. We also found that the system spends exponentially longer time to acquire Facebook check-in data when the number of check-ins increases. Two variables are examined in Figure 3: execution time and the number of years of check-ins. Figure 3 (A) and (B) demonstrate the user preference prediction precision using personal data and Facebook friend data, respectively. The results show that the ranking accuracies in both line charts become higher when the number of years increases. This result indicates that more check-in data produces higher prediction accuracy. Both figures illustrate the results for determining the optimal value of years of check-ins and execution time. The preference prediction precisions of both cases are increased exponentially from 1 to 5 years and remained steady afterwards. This is because five years of data are sufficient to analyze user preference. Although more information are added into the analysis model, they have small effect to the precision value and this makes the system stabilizes at five years of check-ins. Figure 3 (A) shows the highest preference prediction precision of 81.78% is achieved after 14.95 seconds to execute individual data. Figure 3 (B) indicates that the highest preference prediction precision of 73.52% took 65.43 seconds when using friend data as this data are not replicated in local database. The system will acquire data from friends only when user data is not adequate and, thus, the execution time is high (up to one minute).

B. Optimal Number of Friends

Not all of friends on Facebook are close friends. Using all information of friends on Facebook can greatly increase time for data extraction and could be noise for the user model. Thus, close friends data are more beneficial for the RS system. This experiment aims to study how much data from close friends are needed for the recommendation. To conduct this experiment, close friends are identified by the algorithm ay-fb-friend-rank [5] described in section II (A). We also studied the recommendation accuracy by varying the number of close friends used to collect the check-in data. The average precision and execution time are compared to the results from using data of random sample of friends (any friends). Based upon the results in Figure 4, five of close friends can obtain the highest average accuracy of all categories ranking (79.42%) and consume the lowest execution time (75 seconds). A greater number of close friends seems to provide a higher accuracy of recommendation. However, too many friends can exponentially increase the execution time, whereas the recommendation performance remains steady. In other words, the recommendation performances do not change although more information of friends is collected. In the contrary, the average precision obtained from using any friend data is lower than using close friend data because data from any friends does not effectively represent user interest and can confuse the analysis model.

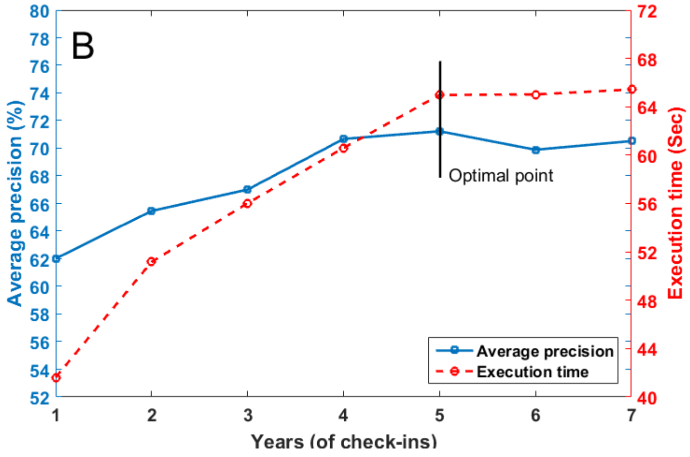
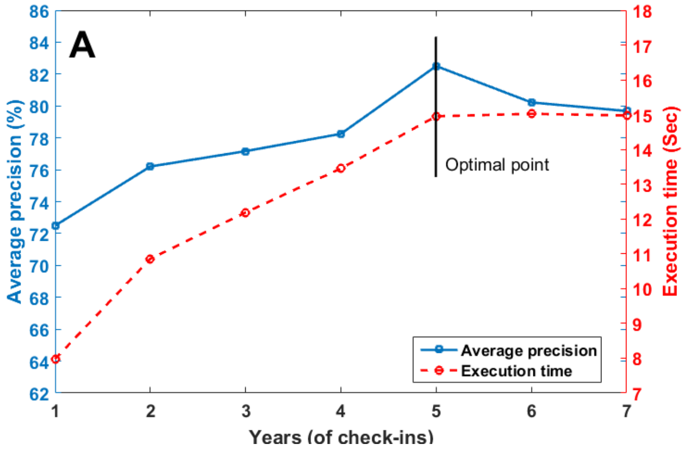


Figure 3: The average precision and execution time for various numbers of years of (A) individual data and (B) friends' data.

The result in Figure 4 indicates that five closest friends are the optimal value, which allows the system to produce high-performance recommendations of relevant attractions for tourists and do not overload the RS. This demonstrates the potential of the exploitation of Facebook friend data for the improvement of personalized recommendation services when no data are available, such as in cold-start situation. This finding is practically important because it can be used as a guideline for other RSs to consider the appropriate amount of data to be used for user-interest analysis carefully, as it affects the recommendation performance and processing time.

Based on the above discussion, the presented approach here could possibly spend significantly less time compared with conventional methods [4], [12], [13] that exploit all user data from tags and cross-domain social networks.



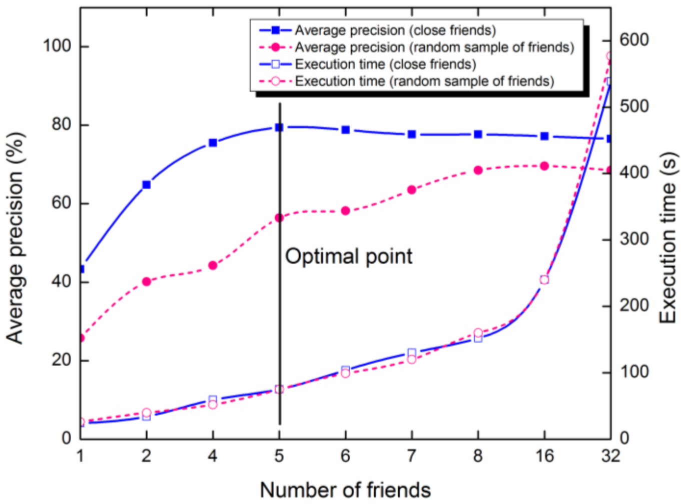


Figure 4: The average accuracy and execution time using different numbers of close friends.

IV. Conclusions

This research proposed an approach to solve the cold start problem of recommendation system using check-in information extracted from Facebook services in the domain of tourism. This information is useful for the analysis of user attraction preference and significantly benefits tourism industries. Although user's check-in data from social network is used for personalized trip recommendation (*PTR*) introduced by Lu *et al.* [14] using *Parallel Trip-Mine* algorithm, PTIS differs from *PTR* and state-of-the-art approaches presented in the literature by overcoming cold-start problem by collecting information from individual users and friends available on Facebook. Here, close friends are detected based on three parameters: affinity score, edge weight, and time decay. The PTIS uses close friends' information to identify attractions in which the target user may be interested. As a result, the PTIS can also serve as a solution for the cold-start user problem, which is the weakness of the state-of-the-art approaches. In addition, we also discovered that too much information of a certain number of close friends can effectively reduce the user-interest extraction and processing time while providing the same recommendation accuracy.

On direction of our future work is to incorporate context awareness into the presented framework. Context can be any information regarding the tourism situation in which a user experiences an attraction (e.g., location, time, or weather). Another direction is that user interest change detection technique could be benefit to PTIS because user preferences are not static. Adding those functions into our frameworks can lead the system to be more reliable and practical.

Acknowledgment

This research has been supported in part by Naresuan University, Thailand. Grant no. R2559C228.

Author contributions

The following authors collected data for the experiment: W. Juraphanthong and K. Kesorn. The following authors conceived, designed the experiments, and wrote the paper: A. Salaiwarakul and K. Kesorn. The following authors were involved in the discussions and analysis plans for the paper from its inception, including the idea of the data analysis: A. Salaiwarakul and K. Kesorn.

References

[1] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, "Facing the cold start problem in recommender systems," *Expert Syst. Appl.*, vol. 41, no. 4, Part 2, pp. 2065–2073, 2014.

[2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, 2005.

[3] V. A. Rohani, Z. M. Kasirun, S. Kumar, and S. Shamshirband, "An effective recommender algorithm for cold-start problem in academic social networks," *Math. Probl. Eng.*, vol. 2014, p. e123726, 2014.

[4] B. Shapira, L. Rokach, and S. Freilikhman, "Facebook single and cross domain data for recommendation systems," *User Model. User-Adapt. Interact.*, vol. 23, no. 2–3, pp. 211–247, 2012.

[5] K. Kuizinas, "Facebook-friend-rank," *GitHub*, 2012. [Online]. Available: https://github.com/gajus/facebook-friend-rank.

[6] J. Widman, "EdgeRank," *EdgeRank*, 2014. [Online]. Available: http://edgerank.net/.

[7] J. Yuan, H. Zhang, and J. Ni, "A new ontology-based user modeling method for personalized recommendation," in *2010 3rd International Conference on Computer Science and Information Technology*, 2010, vol. 4, pp. 363–367.

[8] L. Razmerita, "Ontology-based user modeling," in *Ontologies*, R. Sharman, R. Kishore, and R. Ramesh, Eds. Springer US, 2007, pp. 635–664.

[9] K. Kesorn, Z. Liang, and S. Poslad, "Use of granularity and coverage in a user profile model to personalise visual content retrieval," in *Second International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services, 2009. CENTRIC '09*, 2009, pp. 79–84.

[10] K. Kabassi, "Personalizing recommendations for tourists," *Telemat. Inform.*, vol. 27, no. 1, pp. 51–66, 2010.

[11] Y. Yusof and Z. Mustaffa, "Dengue outbreak prediction: A least squares support vector machines approach," *Int. J. Comput. Theory Eng.*, vol. 3, no. 4, pp. 489–493, 2011.

[12] F. G. Davoodi and O. Fatemi, "Tag based recommender system for social bookmarking sites," in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2012, pp. 934–940.

[13] H. Kumar, S. Lee, and H.-G. Kim, "Exploiting social bookmarking services to build clustered user interest profile for personalized search," *Inf. Sci.*, vol. 281, pp. 399–417, 2014.

[14] E. H.-C. Lu, C.-Y. Chen, and V. S. Tseng, "Personalized trip recommendation with multiple constraints by mining user check-in behaviors," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, 2012, pp. 209–218.