Personalized Hotel Recommendation based on Social Networks

Shaowu Liu and Gang Li

Abstract

Recommender systems have become an important tool for users to identify interesting items as well as for businesses to promote their products to the right users. With the rapid development of social networks, travelers start to seek recommendations and advises from websites like TripAdvisor and Yelp. While travelers are willing to share their opinions on social networks, this provides an opportunity for hospitality businesses to learn their customers' preferences. Given these preferences data, recent advances in machine learning research has made it possible to build automatic recommender systems that can generate hotel recommendations tailored for each traveler. This chapter introduces the basic concepts and tools for creating hotel recommender systems.

Keywords: Recommender Systems, Hotel Recommendation, Collaborative Filtering

1 INTRODUCTION

Recommender Systems (RecSys) aim to suggest items (hotels, books, movies, tourism attractions, etc.) that are potentially to be liked by users. To identify the appropriate items, RecSys use various sources of information, such as the historical ratings given by the users and the content of the items. RecSys were originally designed for users with insufficient personal experience or with limited knowledge on the items. However, with the rapid expansion of Web 2.0 and e-commerce, overwhelming number of items are offered, and every user can be benefited from RecSys.

Hotel recommendation is a well-studied topic in hospitality research (Chen and Chuang, 2016; Jannach et al, 2012). However, most travelers received similar recommendations via

static methods, such as newspapers and television. Advances of Internet has made hotel recommendation in a more interactive form, where travelers can now read reviews and recommendations shared by other travelers on social network, such as Twitter, TripAdvisor and Yelp. However, in all of these recommendation scenarios, travelers receive the same recommendation without personalization. For example, a traveler with limited budget may still be recommended with an expensive hotel because of its high average rating. Considering there are thousands of hotels in a popular destination, it is impractical for travelers to find out the hotel they really need by simply sorting the hotels via a criterion. Consequently, personalized hotel recommendation is needed to identify a small set of hotels what are potentially to be liked by travelers.

Over the last decade there have been rapid advances in RecSys, from both academia and industry (Bennett and Lanning, 2007; Knijnenburg et al, 2012; Li et al, 2015). Numerous recommendation techniques have been proposed to achieve personalized recommendation. However, have been limited work on personalized hotel recommendation (Garbers et al, 2006; Saga et al, 2008; Yu and Chang, 2009; Xiong and Geng, 2010) due to issues such as cold-start and non-rating data. This paper aims to identify issues presented in personalized hotel recommendation and review recommendation techniques in the context of hospitality.

2 PERSONALIZED HOTEL RECOMMENDATION FOR INDIVIDUALS

Hotel recommendation is not a new thing, and it is overlapped with hotel selection. Traditionally, the preferences of travelers are unknown or known to a limited extend, thus all travelers receive similar recommendation lists by measuring the overall quality of hotels. Fortunately, social network has made it possible to get a better understanding of travelers by analyzing information they shared on social networks, such as reviews, ratings, profiles, and social connections. With this rich information available, personalized hotel recommendation becomes possible. In this section, we review how personalized hotel recommender systems can be built using information shared over social networks.

2.1 Recommendation using Explicit Feedback

Social network websites such as TripAdvisor and Yelp provide travelers a virtual place to share their opinions on hotels. While other options are possible, ratings are the most commonly used format of review. For example, TripAdvisor allows travelers to rate a 1-5 star on the hotel, and optionally to different dimensions of the hotel, such as *cleanness*, *location*, and *service*. Despite of the popular star ratings, some websites tend to use other formats, such thumbs up and thumbs down in Facebook. These kinds of feedback provided by travelers are call *Explicit Feedback*, where the travelers explicitly tell us whether they like or dislike the hotel. In general, explicit feedback-based recommender systems can be categorized into content-based filtering and collaborative filtering.

2.1.1 Content-based Filtering

Content-based methods (Lops et al, 2011; Pazzani and Billsus, 2007) generate recommendations by exploiting regularities in the item content. For example, *actors*, *directors*, and *genres* can be extracted as content of movies. In the context of hotel recommendation, the content could be *location*, *price*, *star rating*, etc. To make recommendations for a traveler u, we just need to find out which hotels are similar to the hotels the traveler liked before, i.e., highly rated by traveler u. The similarity between two hotels t_x and t_y can be computed by popular measures such as Pearson Correlation Coefficient (PCC) and Vector Space Similarity.

Despite of its simplicity, content-based methods have limitations. Firstly, it can be difficult to define features or extract content from some hotels. Secondly, travelers will always be recommended with hotel that are highly similar to the hotels he/she liked, which leads to the lacking of diversity (Bradley and Smyth, 2001) and a potentially better hotel may never be recommended.

2.1.2 Collaborative Filtering

Collaborative Filtering methods generate recommendations by analyzing preferences provided by travelers, e.g., ratings. One of the most popular and accurate CF method is Matrix Factorization (MF) (Koren et al, 2009). This approach discovers the latent factor spaces shared between travelers and hotels, where the latent factors can be used to describe both the taste of travelers and the characteristics of hotels. The attractiveness of a hotel to a traveler is then measured by the inner product of their latent feature vectors.

Formally, each traveler u is associated with a latent feature vector $p_u \in \mathbb{R}^k$ and each item i is associated with a latent feature vector $q_i \in \mathbb{R}^k$, where k is the number of factors. The aim of MF is then to estimate $\hat{r}_{ui} = b_{ui} + p_u^T q_i$ such that $\hat{r}_{ui} \cong r_{ui}$. The bias term $b_{ui} = \mu + b_u + b_i$ takes the biases into consideration, where μ is the overall average rating, b_u is the traveler bias, and b_i is the hotel bias. The latent feature vectors are learned by minimizing regularized squared error with respect to all known preferences:

$$\min_{p_u, q_u \in \mathbb{R}^k} \sum_{r_{ui} \in R} (r_{ui} - b_{ui} - p_u^T q_i)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2)$$

λ is the regularization coefficient. The optimization can be done with Stochastic Gradient Descent for the favor of speed on sparse data, or with Alternating Least Squares for the favor of parallelization on dense data.

2.2 Recommendation using Implicit Feedback

Not all users are willing to rate their preferences, where collecting feedbacks implicitly delivers a more user-friendly RecSys. Examples of implicit feedback include the time a user stayed on a webpage, the number of clicks a user performed on an item, and location information of users. The importance of implicit feedback has been recognized recently, and it provides an opportunity to utilize the vast amount of implicit data that have already been collected over the years, such as activity logs. In this section, we review implicit feedback-based recommender systems in the context of hotel recommendation.

2.2.1 Relative Preference-based Filtering

A preference relation (PR) encodes user preferences in form of pairwise ordering between items, i.e., is item X is better than item Y? This representation is a useful alternative to explicit ratings as it can be inferred from implicit data. For example, the PR over two Web pages can be inferred by the stayed time, and consequently applies to the displayed hotels.

PR is formally defined as follows. Let $\mathrm{U} = \{\mathrm{u}\}^n$ and $\mathrm{I} = \{\mathrm{i}\}^m$ denote the set of n travelers and m hotels, respectively. The PR of a traveler $\mathrm{u} \in \mathrm{U}$ between hotels i and j is encoded as π_{uij}, which indicates the strength of traveler u's PR for the ordered hotel pair (i, j). A higher value of π_{uij} indicates a stronger preference on the first hotel over the second hotel (Desarkar et al, 2012; Liu et al, 2015):

$$\pi_{uij} = \begin{cases} \left(\frac{2}{3}, 1\right] \; if \; i \succ j \; (\, u \; prefers \; i \; over \; j) \\ \left[\frac{1}{3}, \frac{2}{3}\right] \; if \; i \simeq j \; (\, equally \; preferable) \\ \left[0, \frac{1}{3}\right) if \; i \prec j \; (\, u \; prefers \; j \; over \; i) \end{cases}$$

The PR can be converted into user-wise preferences over hotels:

$$p_{ui} = \frac{\sum_{j \in I_u} \left[\!\left[\pi_{uij} > \frac{2}{3} \right]\!\right] - \sum_{j \in I_u} \left[\!\left[\pi_{uij} < \frac{1}{3} \right]\!\right]}{|\prod_{ui}|}$$

Where $[\![\cdot]\!]$ gives 1 for true and 0 for false, and $\prod_{ui}$ is the set of traveler u's PR related t hotel i.

Once the user-wise preferences are computed from implicit feedback, they can be set as input for model-based collaborative filtering methods (Brun et al, 2010; Desarkar et al, 2012; Liu et al, 2015).

2.2.2 Text-based Filtering

Online reviews may contain both ratings and text-based comments. While ratings are easy to process, it remains a challenge problem of extracting useful information from textual reviews. However, textual reviews can be particularly useful when travelers do not provide enough ratings. For example, TripAdvisor allows travelers to rate hotels on several optional dimensions such as *cleanness* and *service*. When the rating of a dimension is missing from the traveler, it can be filled by extracting the traveler's opinion from textual reviews. Extracting opinions from text is the task of sentiment analysis and opinion mining (Liu 2012), which can be further divided into two sub-tasks: topic identification and opinion extraction.

In general, the first step is to identify topics from text. For example, a review comments may contain many sentences, and a method is required to classify which topic a sentence belongs to, e.g., *cleanness*. This can be done using simple keywords matching method (Liu et al, 2013) or advanced techniques such as topic models (Mei et al, 2007).

Once the topics are identified, the second task is to extract positivity, negativity, and subjectivity opinions from associated text. One method is to look up words and/or phrases into sentiment dictionaries, such as SentiWordNet 3.0 (Baccianella et al, 2010). Having the opinions extracted, missing ratings can be filled and a denser dataset is obtained for better recommendation performance.

2.3 Evaluation of Hotel Recommender Systems

The evaluation metrics are essential for building successful recommender systems. Efforts have been done to identify the proper way of measuring quality of recommendations. This section reviews common evaluation metrics for hotel recommender systems.

2.3.1 Accuracy Metrics

Two popular metrics are Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), which measure the differences between the predicted preferences and true preferences. Let N be the number of unrated items by user u_a, and $\hat{r}_i$ be the predicted rating of item t_i, the definition of MAE and RMSE are as follows:

$$\text{MAE} = \frac{\sum_{a,i} |\hat{r}_{a,i} - r_{a,i}|}{\text{N}}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{a,i} (\hat{r}_{a,i} - r_{a,i})^2}{\text{N}}}$$

2.3.2 Diversity

Traditionally, the evaluation of RecSys is mainly based on accuracy metrics such as RMSE. However, the accuracy metrics fail to evaluate some properties of the hotels other than the preferences, such as Serendipity (Ge et al, 2010) and Diversity (Zhou et al, 2008). For example, a hotel recommendation list should contain both budget hotels and luxury hotels even if a traveler prefers budget hotels in most cases.

One diversity metric is Personalization, in which the uniqueness of each user's recommendation list is measured. Personalization refers the inter-user diversity (Zhou et al, 2008):

$$\text{Personalization} = \frac{2}{m(m-1)} \sum_{x \neq y} \left(1 - \frac{|L_k(u_x) \cap L_k(u_y)|}{L_k(u_x)}\right)$$

Where m is the number of users, and $\left(1 - \frac{|L_k(u_x) \cap L_k(u_y)|}{L_k(u_x)}\right)$ is the Hamming distance between recommendation lists $L_k(u_x)$ and $L_k(u_y)$.

2.3.3 Coverage

Coverage refers to the percentage of hotels out of all hotels a RecSys can recommend. This metric is based on the observation that some hotels may not have the chance to be recommended to any traveler if it is not popular, e.g., a new hotel.

Let N be the length of recommendation list, L_d be the number of distinct hotels in all Top-N recommendation lists, and L be the number of distinct hotels in all recommendation lists. The N-dependent coverage is defined as (Ge et al, 2010):

$$\mathrm{Coverage(N)} = \mathrm{L}_d / L$$

A low coverage means the RecSys can only make recommendations on a small number of distinct hotels, in other words, it always recommends the popular hotels. Note that RecSys with high coverage implies higher diversity (Lü and Liu, 2011).

2.3.4 Stability

Stability measures consistency of recommendations for the same traveler (Adomavicius and Zhang, 2012). The recommendations generated by a stable RecSys should be similar after some new preferences are added. For example, the first recommendation of an unstable RecSys predicts hotel X as 5-star and hotel Y as 1-star. Then the traveler stayed in hotel X and rated it as 5-star. With this new preference added to the preferences data, an unstable RecSys may generate the second recommendation that predicts hotel Y as 5-star. The 5-star hotel Y which was 1-star, may lead to user confusion and lower the trust of the RecSys. The Stability property has been studied in detail in (Adomavicius and Zhang, 2012).

3 PERSONALIZED HOTEL RECOMMENDATION FOR GROUPS

In real-world applications, there are many scenarios where recommendations are made for a group of travelers, such as holiday packages (McCarthy et al, 2007) and tourist

promotions (Garcia et al, 2009). Group Recommender Systems (G-RecSys) focuses on making recommendations that fit the needs of a group of travelers, instead of individuals. In classic RecSys, the goal is to maximize the satisfaction of a single traveler. However, G-RecSys need to make trad-off among travelers in the group, where the optimal recommendations that satisfy everyone often do not exist.

Recent developments in social networks and interactive media (e.g. interactive TV) have further linked users into groups (Gartrell et al, 2010; Vasuki et al, 2010; Yu et al, 2006; Jameson and Smyth, 2007; Masthoff, 2011), and therefore heightened the need for G-RecSys. However, personalized G-RecSys have only been discussed in limited literature comparing to classic RecSys, and this is particularly true in the context of hospitality. A few survey papers have tried to summarize related works. For example:

(1) The influential survey by Jameson and Smyth (2007) divided group recommendation into four sub-tasks: Group Preference Specification, Group Recommendation Generation, Explaining Recommendations, and Achieving Consensus. Descriptions are given on how existing G-RecSys handle these tasks.
(2) Boratto and Carta (2010) classified user groups into four types: Established Group, Occasional Group, Random Group, and Automatically Identified Group. Existing G-RecSys are examined with focuses on how the type of group affects the design of G-RecSys.
(3) More recently, Masthoff (2011) surveyed techniques used in the Group Recommendation Generation sub-task. Eleven aggregation strategies inspired by Social Choice Theory are summarized with discussions on existing G-RecSys.

Current G-RecSys research mainly focus on answering the following four questions: 1) How to collect and represent preferences? 2) How to generate recommendations by

aggregating preferences of individuals? 3) How to explain the recommendations? 4) How to help group users arriving at a final decision?

3.1 Group Recommendation Generation

Group Recommendation Generation refers to the process of aggregating group users' preferences and making recommendations based on the aggregated preferences. Regardless of preference specification, individual users' preferences have to be aggregated in some way, and identifying the proper aggregation approach has been the main focus in literature (Jameson and Smyth, 2007; Arrow 2012). In general, there are three approaches to generate group recommendations, and all require preference aggregations (Jameson and Smyth, 2007):

(1) Merging Recommendations for Individuals: In this approach, the classic RecSys will be applied to make recommendations for individuals. The recommendations for a group is then computed by merging the recommended items for each individual in the group. The merging is controlled by a selected aggregation function and in the simplest case the items with highest predicted ratings for individuals are selected.

(2) Aggregating Preferences of Individuals: This approach also relies on the individuals' ratings predicted by the classic RecSys. The difference is that instead of making a list of recommendations for each individual, the ratings for each item is aggregated. In other words, each item received a rating aggregated from preferences of all group users. The group recommendation is made by selecting the items with highest ratings.

(3) Constructing Group Preference Models: This approach does not require predictions of ratings for individual users. Instead, the known preferences of individual users are aggregated into a single profile for the whole group. After the aggregation, the group looks no different from a normal user, and recommendations are made for this group using classic RecSys.

Basically, G-RecSys either aggregate preferences of individuals or construct a group preference model. The main advantage of Group Preference Models over preference aggregations is the privacy benefits. When users' preferences are aggregated into a group preference model, the individual user's preferences are hid- den. However, preference aggregation methods can make better recommendations in some cases. For example, items recommended by preference aggregation approaches won't be disliked by all group users, where it is possible, though unlikely, that no group user likes the items recommended by Group Preference Models. No matter which approach is selected, the main issue involved is how to perform aggregation. Most aggregation methods discussed in existing surveys are inspired by strategies from Social Choice Theory (Arrow 2012). For example, the Maximizing Average strategy will recommend item that can achieve the highest average rating from group members. On the other hand, the Minimizing Misery will discard items that are very disliked by any group member even if the average rating is high. This kind of strategies are very intuitive but selecting which one to use is a manual process. The choice of aggregation methods is often left as an open question or very basic ones are used (Amer-Yahia et al, 2009). However, a lot of established aggregation methods have been developed in communities other than RecSys and Social Choice Theory, such as Fuzzy Integrals (Beliakov et al, 2007). These techniques are powerful tools to aggregate data, and are often less context dependent.

3.2 Explaining Recommendations

Explaining Recommendations (McSherry, 2005; Knijnenburg et al, 2012) is the task of making the recommendation process more transparent to the users, i.e. why these items are recommended? how confident the recommendations will be liked? For example, a RecSys could make the following explanation (O'Donovan and Smyth, 2005): "the items are recommended to you because they have been successfully recommended to users A, B, and C who are similar to you. In addition, we have made X, Y, and Z times recommendations

to them in the past, which received P, Q, and R likes". In the context of group recommendation, the Explaining Recommendations task refers to make group users fully understand the recommendations. However, the primary goal of explanation is not to convince the users about the proposed recommendations, but helping the users to understand other group users' feelings about the recommendations. This process will help the group users to adjust the proposed recommendations to arrive a final decision. Unlike classic RSs, debates and negotiations are often necessary for group users, and this calls for understanding of not only the pros but also the cons of the proposed recommendations. While existing explanation approaches focus on determining how good the recommendation is for the user, it is now desirable to know how bad the recommendation is for each group user.

3.3 Achieving Consensus

The proposed recommendations can be a promising solution but may eventually be rejected by the group. Making the final decision is a complex process that may involve extensive debate and negotiations. Typical G-RecSys assume group users are independent and consider each user equally. Technically, G-RecSys is able to identify the recommendations that maximize the overall satisfaction of the group, however, the true maximized satisfaction may not be achieved when interactions exist among group users. For example, when recommending travel destination for a family, the recommended destination may maximize the average satisfaction of all family members. However, the parents may prefer another destination over their favorites because they care about the children' satisfaction, but on the other hand, the children may not consider their parents' satisfaction too much. In this case, one of the children' favorite destinations that not disliked by the parents may be the final decision. Ideally, G-RecSys should take such in-group interactions into consideration, either prior the recommendation generation or make adjustment after received feedback of proposed recommendations. Considering user interactions in recommendation generation has been studied by Amer-Yahia et al. (2009), where a consensus function is defined to maximizing item relevance and minimizing disagreements

between group users. However, modeling complex user interactions remain an unsolved research problem. Another way to consider user interactions is to make adjustment by evaluating feedback of proposed recommendations. This kind of process is called Reinforcement Learning, and has been applied in the context of classic RecSys (Taghipour et al, 2007; Mahmood and Ricci, 2009).

4 RECOMMENDER SYSTEMS SOFTWARE PACKAGES

Though many companies have implemented their own recommender systems according to their specific business needs, there are still many free/open source recommender system software available. In this section, we review some popular software packages for practitioners to build their hotel recommender systems.

4.1 MyMediaLite

MyMediaLite (http://mymedialite.net/) is a recommender system library for the Microsoft .NET platform, and runs on Linux and Mac OS X through the Mono platform. It implements common RecSys algorithms to build models from both explicit ratings and implicit feedback. The software is free and open source, and can be used, modified, and distributed under the terms of the GNU General Public License (GPL).

4.2 Apache Mahout

Apache Mahout (http://mahout.apache.org/) is a scalable machine learning library which implemented a few standard recommender system algorithms. This software is particularly useful for building recommender systems on large amount of data, e.g., 100 million records. The software is implemented in Java programming language and is free/open source under Apache License.

4.3 Recommenderlab

Recommenderlab (https://cran.r-project.org/web/packages/recommenderlab/index.html) is a package for R programming language. It provides a research infrastructure to test and develop recommender algorithms including UBCF, IBCF, FunkSVD and association rule-based algorithms. The software is free/open source under GPL-2 license.

4.4 Easyrec

Easyrec (http://easyrec.org/) is a free/open source software that can easily integrate recommender systems into website though plugins and javascript code. This is particularly useful for hotels who wants to add simple recommendation functions to their website with limited resources.

4.5 waffles_recommend

waffles_recommend (http://uaf46365.ddns.uark.edu/waffles/command/recommend.html) is a command-line tool for predicting missing values in incomplete data, or for testing collaborative filtering recommendation systems. It provides simple recommender system algorithms and is computationally efficient.

4.6 LensKit

LensKit (http://lenskit.org/) is a software package implements many popular collaborative filtering algorithms and provide a set of tools to benchmark them. The software is implemented in Java programming language. The software is free and open source under General Public License (GNU).

4.7 GraphLab (Turi)

GraphLab/Turi (https://turi.com/) is a sophisticated machine learning platform. It implements recommender system algorithms and provide commercial support. However, one-year free subscription is available for academic use.

5 CONCLUSIONS

This chapter aims to present the start of the art in recommender systems for the purpose of hotel recommendation. This has included recommendation techniques using explicit feedback, such as ratings. We also reviewed recommendation techniques using implicit feedback, such as *clicks* and *page views*, which is gaining popularity in recent years. To evaluate recommender systems, we reviewed commonly used metrics, including accuracy

metrics, diversity, coverage, and stability. In addition, we provide a list of free and open source software packages for practitioners to create their own recommender systems.

There has been extensive research on the topic of recommender systems, some of which have been applied to hotel recommendation. As this chapter provides only an introduction to this topic, we recommend a list of books and papers under *Further Readings* for readers.

6 FURTHER READINGS

- Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. Knowledge-Based Systems, 46, 109-132. (An extensive survey of recommender systems)
- Chen, L., Chen, G., & Wang, F. (2015). Recommender systems based on user reviews: the state of the art. User Modeling and User-Adapted Interaction, 25(2), 99-154. (Create recommender systems from user reviews)
- Gavalas, D., Konstantopoulos, C., Mastakas, K., & Pantziou, G. (2014). Mobile recommender systems in tourism. Journal of Network and Computer Applications, 39, 319-333. (Recommender systems under mobile environment)
- Liu, B. (2012). Sentiment analysis and opinion mining. Synthesis lectures on human language technologies, 5(1), 1-167. (Explaining how to extract opinions from text-based reviews)
- Lü, L., Medo, M., Yeung, C. H., Zhang, Y. C., Zhang, Z. K., & Zhou, T. (2012). Recommender systems. Physics Reports, 519(1), 1-49. (An extensive survey of recommender systems)
- Park, D. H., Kim, H. K., Choi, I. Y., & Kim, J. K. (2012). A literature review and classification of recommender systems research. Expert Systems with Applications, 39(11), 10059-10072. (An extensive survey of recommender systems)

- Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook (pp. 1-35). Springer US. (This handbook covers most topics of recommender systems)

References

Adomavicius, G., & Zhang, J. (2012). Stability of recommendation algorithms. ACM Transactions on Information Systems (TOIS), 30(4), 23.

Amer-Yahia, S., Roy, S. B., Chawlat, A., Das, G., & Yu, C. (2009). Group recommendation: Semantics and efficiency. Proceedings of the VLDB Endowment, 2(1), 754-765.

Arrow, K. J. (2012). Social choice and individual values (Vol. 12). Yale university press.

Baccianella, S., Esuli, A., & Sebastiani, F. (2010, May). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In LREC (Vol. 10, pp. 2200-2204).

Beliakov, G., Pradera, A., & Calvo, T. (2007). Aggregation functions: A guide for practitioners (Vol. 221). Heidelberg: Springer.

Bennett, J., & Lanning, S. (2007, August). The netflix prize. In Proceedings of KDD cup and workshop (Vol. 2007, p. 35).

Bradley, K., & Smyth, B. (2001). Improving recommendation diversity. In Proceedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science, Maynooth, Ireland (pp. 85-94).

Brun, A., Hamad, A., Buffet, O., & Boyer, A. (2010, September). Towards preference relations in recommender systems. In Preference Learning (PL 2010) ECML/PKDD 2010 Workshop.

Chen, T., & Chuang, Y. H. (2016). Fuzzy and nonlinear programming approach for optimizing the performance of ubiquitous hotel recommendation. Journal of Ambient Intelligence and Humanized Computing, 1-10.

Desarkar, M. S., Saxena, R., & Sarkar, S. (2012, July). Preference relation based matrix factorization for recommender systems. In International Conference on User Modeling, Adaptation, and Personalization (pp. 63-75). Springer Berlin Heidelberg.

Garbers, J., Niemann, M., & Mochol, M. (2006). A personalized hotel selection engine. In Proceedings of the third European Semantic Web Conference.

Garcia, I., Sebastia, L., Onaindia, E., & Guzman, C. (2009, September). A group recommender system for tourist activities. In International Conference on Electronic Commerce and Web Technologies (pp. 26-37). Springer Berlin Heidelberg.

Gartrell, M., Xing, X., Lv, Q., Beach, A., Han, R., Mishra, S., & Seada, K. (2010, November). Enhancing group recommendation by incorporating social relationship interactions. In Proceedings of the 16th ACM international conference on Supporting group work (pp. 97-106). ACM.

Ge, M., Delgado-Battenfeld, C., & Jannach, D. (2010, September). Beyond accuracy: evaluating recommender systems by coverage and serendipity. In Proceedings of the fourth ACM conference on Recommender systems (pp. 257-260). ACM.

Jameson, A., & Smyth, B. (2007). Recommendation to groups. In The adaptive web (pp. 596-627). Springer Berlin Heidelberg.

Jannach, D., Gedikli, F., Karakaya, Z., & Juwig, O. (2012). Recommending hotels based on multi-dimensional customer ratings. Information and Communication Technologies in Tourism. Springer-Verlag/Wien.

Knijnenburg, B. P., Willemsen, M. C., Gantner, Z., Soncu, H., & Newell, C. (2012). Explaining the user experience of recommender systems. User Modeling and User-Adapted Interaction, 22(4-5), 441-504.

Knijnenburg, B. P., Willemsen, M. C., Gantner, Z., Soncu, H., & Newell, C. (2012). Explaining the user experience of recommender systems. User Modeling and User-Adapted Interaction, 22(4-5), 441-504.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37.

Li, X., Xu, G., Chen, E., & Zong, Y. (2015). Learning recency based comparative choice towards point-of-interest recommendation. Expert Systems with Applications, 42(9), 4274-4283.

Liu, B. (2012). Sentiment analysis and opinion mining. Synthesis lectures on human language technologies, 5(1), 1-167.

Liu, S., Law, R., Rong, J., Li, G., & Hall, J. (2013). Analyzing changes in hotel customers' expectations by trip mode. International Journal of Hospitality Management, 34, 359-371.

Liu, S., Li, G., Tran, T., & Jiang, Y. (2015). Preference Relation-based Markov Random Fields for Recommender Systems. In Proceedings of The 7th Asian Conference on Machine Learning (pp. 157-172).

Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In Recommender systems handbook (pp. 73-105). Springer US.

Lü, L., & Liu, W. (2011). Information filtering via preferential diffusion. Physical Review E, 83(6), 066119.

Mahmood, T., & Ricci, F. (2009, June). Improving recommender systems with adaptive conversational strategies. In Proceedings of the 20th ACM conference on Hypertext and hypermedia (pp. 73-82). ACM.

Masthoff, J. (2011). Group recommender systems: Combining individual models. In Recommender systems handbook (pp. 677-702). Springer US.

McCarthy, K., McGinty, L., & Smyth, B. (2007, August). Case-based group recommendation: Compromising for success. In International Conference on Case-Based Reasoning (pp. 299-313). Springer Berlin Heidelberg.

McSherry, D. (2005). Explanation in recommender systems. Artificial Intelligence Review, 24(2), 179-197.

Mei, Q., Ling, X., Wondra, M., Su, H., & Zhai, C. (2007, May). Topic sentiment mixture: modeling facets and opinions in weblogs. In Proceedings of the 16th international conference on World Wide Web (pp. 171-180). ACM.

O'Donovan, J., & Smyth, B. (2005, January). Trust in recommender systems. In Proceedings of the 10th international conference on Intelligent user interfaces (pp. 167-174). ACM.

Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In The adaptive web (pp. 325-341). Springer Berlin Heidelberg.

Product and User Characteristics. In Management and Service Science (MASS), 2010 International Conference on (pp. 1-5). IEEE.

Saga, R., Hayashi, Y., & Tsuji, H. (2008, October). Hotel recommender system based on user's preference transition. In Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on (pp. 2437-2442). IEEE.

Taghipour, N., Kardan, A., & Ghidary, S. S. (2007, October). Usage-based web recommendations: a reinforcement learning approach. In Proceedings of the 2007 ACM conference on Recommender systems (pp. 113-120). ACM.

Vasuki, V., Natarajan, N., Lu, Z., & Dhillon, I. S. (2010, September). Affiliation recommendation using auxiliary networks. In Proceedings of the fourth ACM conference on Recommender systems (pp. 103-110). ACM.

Xiong, Y. N., & Geng, L. X. (2010, August). Personalized Intelligent Hotel Recommendation System for Online Reservation--A Perspective of

Yu, C. C., & Chang, H. P. (2009, September). Personalized location-based recommendation services for tour planning in mobile tourism applications. In International Conference on Electronic Commerce and Web Technologies (pp. 38-49). Springer Berlin Heidelberg.

Yu, Z., Zhou, X., Hao, Y., & Gu, J. (2006). TV program recommendation for multiple viewers based on user profile merging. User modeling and user-adapted interaction, 16(1), 63-82.

Zhou, T., Jiang, L. L., Su, R. Q., & Zhang, Y. C. (2008). Effect of initial configuration on network-based recommendation. EPL (Europhysics Letters), 81(5), 58004.

BIOGRAPHICAL NOTE

Shaowu Liu is a postdoctoral researcher in the Advanced Analytics Institute at University of Technology Sydney, Australia.

Gang Li is Associate Professor in the School of Information Technology at the Deakin University, Australia.