Hotel Recommendation based on User Preference Analysis

Kai Zhang, Keqiang Wang, Xiaoling Wang, Cheqing Jin, Aoying Zhou

Shanghai Key Laboratory of Trustworthy Computing, Institute for Data Science and Engineering, East China Normal University, Shanghai, China

{xlwang, cqjin, ayzhou}@sei.ecnu.edu.cn

Abstract—**Recommender system offers personalized suggestions by analyzing user preference. However, the performance falls sharply when it encounters sparse data, especially meets a cold start user. Hotel is such kind of goods that suffers a lot from sparsity issue due to extremely low rating frequency. In order to handle these issues, this paper proposes a novel hotel recommendation framework. The main contribution includes: 1) We combine collaboration filtering (CF) with content-based (CBF) method to overcome sparsity issue, while ensuring high accuracy. 2) Travel intents are introduced to provide additional information for user preference analysis. 3) To provide as broad as possible recommendations, diversity techniques are employed. 4) Several experiments are conducted on the real *Ctrip*[1] dataset, the results show that the proposed hybrid framework is competitive against classical approaches.**

Keywords—recommender system; matrix factorization; cold start; text mining; diversity

I. INTRODUCTION

In recent years, urban tourism becomes more and more popular. With the help of E-commerce, it is convenient to choose a sight-seeing destination through Internet. But here comes a typical question for a trip: if we plan to visit the Forbidden City in Beijing for several days, where should we live? Selecting a suitable hotel can be vital for a pleasant trip. In general, this question corresponds to recommending a hotel given a certain destination. However, the past hotel check-in data of an individual may be sparse due to low rating frequency, which means the performance of rating based collaboration filtering technique is poor. Moreover, it's important to serve for newly registered customers who has no hotel booking history [1]. The sparsity issue and the so-called cold start problem are the main challenges for hotel recommendation.

This paper proposes a novel framework to solve these issues. First, CF method is combined with CBF method to overcome sparsity issue, while guaranteeing high enough accuracy. Existing recommender systems basically adopt two kinds of methods: collaborative filtering (CF) or content-based filtering (CBF). CF focuses on finding similar users based on user-item rating matrix, the precision of which has proven to be good but sensitive to sparse data. On the other hand, CBF manages to match the content of item with users' interest, the precision of which is usually not well since it depends on the quality of extracted features. To overcome the drawback of these two methods, an idea of combining them together is then proposed.

Next, we notice that the intent of trip serves as a vital factor when selecting a hotel. For example, those users on a business trip are more concerned about noise while those on a family trip care more about service and facilities. In general, people who have the same intent share similar preference. Such information is easy to acquire and effective for cold start users. For this reason, the travel intent are introduced into the proposed framework for solving cold start problem.

Furthermore, one's preference for hotels tends to be complicated. When recommender system encounters a new user, background knowledge may be insufficient. It is hard to decide whether specific hotel match his/her preference o not. One possible solution is to employ diversity techniques [2], to satisfy one's preference as broad as possible, so that monotony of the recommendation result is avoided.

Our contributions are listed as follows:

- We focus on user preference analysis and solve sparsity issue by integrating CF and CBF.
- The intent of a trip is introduced to solve cold start problem with higher prediction accuracy.
- We use diversity techniques to optimize the hotel recommendation list.
- Experiments show that the proposed hotel recommendation framework outperforms classical approaches.

The rest of the paper is organized below. In Section II, we summarize the existing techniques for recommendation system and possible solutions to handle sparsity issue. Next, the hotel recommendation framework is proposed in Section III. Then, experiments are conducted to evaluate the performance of our framework. Finally, we make a conclusion and point out some future work in Section V.

II. RELATED WORK

Due to the rapid development of E-commerce, personalized recommender system has become hot in recent years. Based on user-item rating matrices, many CF techniques have been proposed [3]. However, extremely sparse data may seriously deteriorate the performance. Matrix factorization is a kind of dimensionality reduction technique to deal with sparsity problem. Some popular Latent Factorization Models have been proven fairly effective, such as Singular Value Decomposition (SVD) [4], Non-negative Matrix Factorization (NMF) [5], Probabilistic Matrix Factorization (PMF) [6] and

[1] http://www.ctrip.com

its Bayesian version [7]. But it is still challenging to deal with newly registered users, which is known as cold-start problem.

One solution to this problem is to seek contents such as reviews for help, leading to hybrid techniques of CF method and CBF method [8]. McAuley et al. mines the review dimension and proposes the hidden factors and hidden topics (HFT) model to link the latent factors with latent topics which are automatically extracted from review contents [9]. TopicMF model is proposed to further improve rating prediction and accuracy by learning topics for each review, which matches better with users' rating behavior [10]. In order to give more explainable recommendation, the explicit factor model (EFM) [11] is proposed. EFM manage to extract explicit item features and user preference based on sentiment analysis, which is the closest approach to ours. In contrast, we pay more attention to the similarity between users and hotels to improve prediction accuracy.

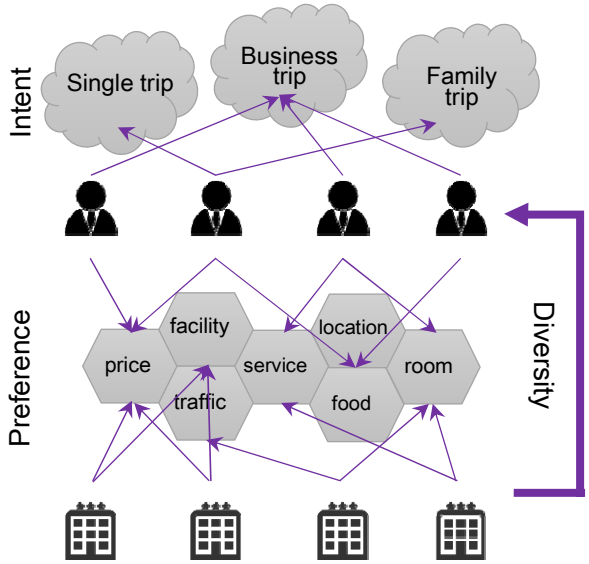

Fig. 1. Hotel Recommendation Framework

Another solution is to look for special features for a specific item, and Lin et al. bring in the version feature and category to recommend apps [12]. The dynamic version description for app update provides valuable content information to match user preference. Levi et al. adopt content-based method and mainly focus on the cold-start problem [13]. They introduce intents into the hotel recommendation. In this paper, we also take the trip intent into hotel recommendation framework as a supplement.

III. Hotel Recommendation Framework

In this section, we describe our hotel recommendation framework, which includes three key parts: user preference, travel intent and diversity, as showed in Fig.1.

In the first step, we determine one's preference on a specific hotel, by predicting missing ratings based on original user-item rating matrices and reviews. The output is a complete rating matrix. In the second step, we use travel intent to modify predicted ratings in this matrix. Then, in the third step, we generate a personalized recommendation list according to one's top-k predicted ratings. Diversity techniques are employed to optimize the recommendation list at last.

A. Preference Factor Model

Latent Factorization Model (LFM) based on Matrix Factorization (MF) has gained great popularity in recent years, due to its good performance and prediction accuracy, such as Singular Value Decomposition (SVD). By learning user-item rating matrices, LFM predicts the missing parts. However, as a kind of CF technique, it bears with the cold-start and sparsity problem. Fortunately, using the idea of CBF method, we can extract features from rich review contents to match a user's preference with a hotel. To overcome the disadvantages of CF method and CBF method, a hybrid preference factor model (PFM) is proposed.

The key step for CBF method is feature extraction, since the quality of prediction mainly depends on those extracted features. In this paper, we use a topic model. One of the well-known topic model is Latent Dirichlet Allocation (LDA), which generates the topic-word and document-topic distribution. With the help of LDA model, we calculate the similarity between two users based on review-preference distribution, and the similarity between two hotels based on review-feature distribution. More specifically, to generate review-preference distribution for a user, we can gather all his/her history reviews together as one document and apply LDA model. Similar, we gather all the reviews for a hotel together as one document and apply LDA model to generate review-feature distribution for this hotel.

We apply Pearson correlation to calculate the similarity:

$$Sim_{i,j} = \frac{\sum_{z\in Z}(r_{i,z}-\overline{r_i})(r_{j,z}-\overline{r_j})}{\sqrt{\sum_{z\in Z}(r_{i,z}-\overline{r_i})^2}\sqrt{\sum_{z\in Z}(r_{j,z}-\overline{r_j})^2}} \quad (1)$$

where the $z \in Z$ summations are generated features in LDA, $r_{i,z}$ denotes the distribution p(z|d) for user i, $\overline{r_i}$ denotes the average distribution of features for user i.

After calculating the Pearson correlation between any pair of users and between any pair of hotels, we obtain two similarity matrixes: S_U and S_V.

We put forward the preference factor model (PFM), as showed in Fig.2 and formulate our object function as:

$$L(U,V) = \|X - UV^T\|^2 + \lambda_1 tr(U^T S_U U) + \lambda_2 \mathrm{tr}(V^T S_V V) + \lambda_3(\|U\|^2 + \|V\|^2) \quad (2)$$

where tr(.) denotes the trace of a matrix, ||.|| denotes the Forbenius norm. λ_1~λ_3 are model parameters, when $\lambda_1 = \lambda_2 = 0$, our model degenerates to the standard SVD.

The first term $\|X - UV^T\|^2$ in (2) controls the loss in matrix factorization. The second term $\lambda_1 tr(U^T S_U U)$ in (2) poses a regularization term on the users, forcing two users as close as possible if they are similar according to their preference. Similarly, the third term $\lambda_2 \mathrm{tr}(V^T S_V V)$ in (2) is used to force two similar hotels as close as possible. The last

term $\lambda_3(\|U\|^2 + \|V\|^2)$ in (2) serves as the regularization over the factorized matrices to prevent over-fitting.

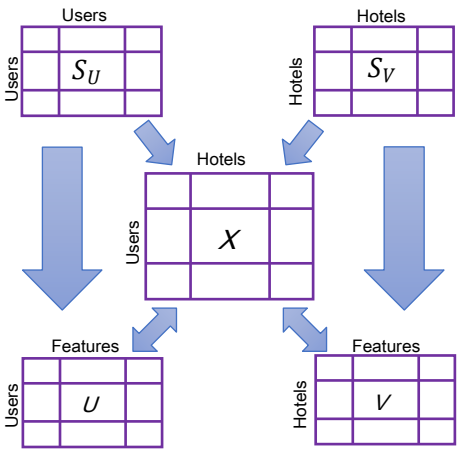

Fig. 2. Preference Factor Model

Note that (2) can be solved by gradient descent, achieving a local optimal solution. More specifically, we have the gradients for each variable as:

$$\nabla_U L = (UV^T - X)V + \lambda_1 S_U U + \lambda_3 U \quad (3)$$
$$\nabla_V L = (UV^T - X)U + \lambda_2 S_V V + \lambda_3 V \quad (4)$$

After having the gradients, we can iteratively minimize the object function by employing gradient descent. The details of the algorithm are as follows:

Algorithm 1 The Gradient Descent Algorithm
Input: Incomplete user-hotel rating matrix X and similarity matrix S_U, S_V.
Output: Complete user-hotel rating matrix X
begin
1: $t = 1$;
2: while $t < T$ and $L_t > L_{t+1}$ do
3: Get the gradients$\nabla_U, \nabla_V$by Eq.(3,4);
4: $U_{t+1} = U_t - \gamma\nabla_U, V_{t+1} = V_t - \gamma\nabla_V$;
5: if($L_t < L_{t+1}$) then break; end if
6: $t = t + 1$;
7: end while
8: return $X = U_t V_t^T$
End

Fig. 3. Algorithm description for PFM

B. Improve Users Predict Rating by Trip Intent

People who have the same trip intent usually share similar hotel preference. Based on this observation, intent can provide additional information for hotel recommendation, which inspires a new solution for cold start problem. It's hard for a recommendation system to predict a user with rare data, especially for a new user. Intent can be specified by a user when using our hotel recommendation framework. According to the dataset used here, the intent of a trip is classified into 6 classes: single, couple, group, family, business and others.

Considering trip intent, our idea is to modify the predicted rating close to the average rating on the same intent class. The sparser data is, the closer rating should be. The predicted rating generated from last step is defined as r_u^h, where h denotes a hotel and u denotes a user. For each hotel, we pre-calculate the mean rating on each intent class, defined as $\bar{r}_i^h$, where h denotes a hotel and i denotes an intent. More specifically, the predicted rating can be modified as follows:

$$r_u^h = \begin{cases} \left(1 - \frac{|r|}{\overline{|r|}}\right) * \bar{r}_i^h + \frac{|r|}{\overline{|r|}} * r_u^h, & |r| < \overline{|r|} \\ r_u^h, & |r| \geq \overline{|r|} \end{cases} \quad (5)$$

where r_u^h denotes the rating predicted by our PFM method during the last step, $\frac{|r|}{\overline{|r|}}$ describes the degree of sparsity, that controls the influence of intent. $|r|_u$ is the number of ratings actually commented by user u, and $|r|_m$ denotes the average rating number per user. The sparser user rating vector is, the smaller $\frac{|r|}{\overline{|r|}}$ will be.

C. Personalized Recommendation

A complete user-hotel rating matrix X is generated by previous steps, which serves as the basis for recommendation. To offer a personalized hotel recommendation list for a specific user, we can simply get all his/her predicted ratings and find top-k hotels with the highest ratings.

By the supplementary information from trip intent, it's been possible to deal with cold start users, but still far from enough to make precise recommendation. Since human preference is usually complicated. In other words, the recommendation result can be monotony and mismatch users' actual preference. Hence, it is better to make the result as broad as possible to satisfy users' unknown preference, which leads to the use of diversity techniques to optimize the recommendation list at the last step. More specifically, we want to make the top-k results less similar with each other when ensuring high enough relevance, which can be regarded as a trade-off between relevance and duplication. We apply maximal marginal relevance (MMR) [14] method to solve this issue:

$$h = arg\ max_{h \in D \setminus S}[\lambda r_u^h / 5 - (1 - \lambda) \max_{h_j \in S} S_V(h, h_j)] \quad (6)$$

where D is the original top-k recommended hotel set, S is the resorted top-k recommended hotel set, S_V refers to the hotels' similarity matrix, and λ denotes the control parameter.

The aim of MMR algorithm (Algorithm 2) is to generate a new resorted ranking set S. MMR employs greedy strategy. It brings the top ranked hotel into the empty set S at first. Then, in every loop, it calculates h according to Eq.(6), picks up the maximal one, and adds it into the reordered set S. During every term, the hotel picked is the least similar one.

Algorithm 2 The Diversity Greedy Algorithm
Input: Predicted user-hotel rating set D and similarity matrix S_V.
Output: Reordered user-hotel rating set S

begin
1: $t = 1$;
2: $S \leftarrow max\ (D)$
3: while $t < |D|$ do
4: $h = arg\ max_{h \in D \setminus S}[\lambda r_u^h/5 - (1-\lambda) \max_{h_j \in S} S_V(h, h_j)]$
5: $S \leftarrow S \cup \{h\}$
6: $t = t + 1$;
7: end while
8: return S
End

Fig. 4. Algorithm description for diversity

IV. EXPERIMENT

In this section, we first introduce the dataset used in our paper, including involved elements and some statistic results. The pre-processing for the reviews are then proposed. After that, the evaluation methods are presented. Finally, we present the experiment results and give corresponding analysis.

A. Settings

The experiment is conducted on real *Ctrip* dataset. *Ctrip* is an E-commence website that provides hotel booking service. The hotel dataset consists of the following elements.

1) Hotel metadata: Hotel metadata consists of a hotel ID, name, longitude, latitude, bottom price, average rating and number of reviews.

2) Review: Besides the content of a review, a review consists of a reviewer ID, hotel ID, rating, intent, check-in date. Intent is the purpose of the trip, which are classified into six categories: single, couple, group, family, business and others.

Some statistics of the dataset is showed in Table I. We randomly separate 80% reviews from dataset as training set, and the rest 20% as testing set, which is commonly adopted in recommendation evaluation.

TABLE I
STATISTICS OF THE CTRIP DATASET

City	#users	#hotels	#reviews	avg.reviews
Beijing	22144	2801	1076721	48.62

Table II lists the number of reviews in each kind of trip intent. There are all six classes of intents: single, couple, group, family, business and others. As we can see, every intent class has at least 50,000 reviews.

TABLE II
STATISTICS OF REVIEW ON TRIP INTENT

single	couple	group	family	business	others
53706	64350	54755	180184	607439	116287

Before experiments, review spam should be removed at first. We mainly discard those meaningless reviews that are too short or have high similarity with others. We set a boundary as less than 5 words to define very short reviews. As for review similarity, there are two directions: based on one hotel or based on one user. The first one refers to those duplicate or near-duplicate reviews on the target hotel, since spam reviewers are used to copying content from nearby. The second one refers to those users that are too lazy to write fresh reviews, they just copy their previous reviews instead. In these two situations, we can remove them by simply calculating their similarities.

B. Evaluation Methodology

To measure the performance of our hotel recommendation framework, we take three CF methods as baselines. The first one is traditional user-based CF approach, which predicts ratings by looking for users who have similar interest. The other two are latent factor models based on matrix factorization: Singular Value Decomposition (SVD) and Bayesian Probabilistic Matrix Factorization (BPMF). We then employ some commonly used evaluation metrics such as Mean Absolute Error (MAE), Root of the Mean Square Error (RMSE) and Asymmetric Measures (ASYMM). They evaluate the error between predicted rating and the real rating in the testing set. They evaluate the error between predicted rating and the real rating in the testing set. The asymmetric loss captures the fact that recommending bad hotels as good hotels is worse than recommending good hotels as bad, in contrast to the standard MAE and RMSE. To validate the effect of trip intent for hotel recommendation, we bring this feature into three different baselines and compare new results with previous ones.

C. Results and Discussions

The number of the latent features for latent factor models is 20. λ_1, λ_2 and λ_3 are all set to 0.1, learning rate is set to 0.05. Table III shows the results without considering the trip intents. We can observe that our PFM behaves better than other three baselines.

TABLE III
PERFORMANCE COMPARISON

Method	MAE	RMSE	ASYMM
UserBsd	0.7062	0.9109	0.5927
SVD	0.7124	0.9066	0.5634
BPMF	0.6944	0.8925	0.5812
PFM	0.69	0.8818	0.5755

Table IV shows the results taking the trip intents into consideration. All the techniques achieve better performance after considering in the trip intents. We can verify that introducing the intent of the trip do improve the prediction accuracy and solves the cold-start problem in hotel recommendation to some extent.

TABLE IV
PERFORMANCE COMPARISON CONSIDERING TRIP INTENT

Method	MAE	RMSE	ASYMM
UserBsd +Intent	0.7044	0.9085	0.5898
SVD +Intent	0.7051	0.9001	0.5523
BPMF +Intent	0.6902	0.8814	0.5634
PFM +Intent	0.6838	0.8783	0.5433

D. Case Study

To take full use of our hotel recommendation framework, we design and implement a demo app based on Android platform. Fig 5 shows a snapshot of our system. The interface provides intuitive recommendation results.

Fig. 5. System for our framework

With the help of our demo system, we are able to verify the effect of personalized hotel recommendation by actual use of some volunteers. Table V shows a top-5 hotel recommendation list for user 205770. As we can see, the explicit preferences of the user are location and traffic since every recommended hotel contains these features.

TABLE V
TOP-5 HOTEL RECOMMENDATION LIST

Hotel Id	Main Features
457224	location; traffic; procedure; surroundings
22244	location; room; traffic; price
43904	location; room; traffic; surroundings
779292	location; traffic; facility; service
225	location; room; traffic; surroundings

TABLE VI
TOP-5 HOTEL RECOMMENDATION LIST CONSIDERING DIVERSITY

Hotel Id	Main Features
457224	location; traffic; procedure; surroundings
22244	location; room; traffic; price
779292	location; traffic; facility; service
43904	location; room; traffic; surroundings
57155	location; room; food; cleanness

After employing diversity techniques, the rank of hotel 43904 and 225 goes down as showed in Table VI The reason is that their main features are very close to hotel 22244, making them repetitive. On the other hand, the ranked up hotel 57155 has some refreshing features which may satisfy user's implicit preference. In this case, the system do meet with the demands of users.

V. CONCLUSION

In this paper, we design a novel hotel recommendation framework. To solve the sparsity issue, we combine latent factor models with content-based method. We bring in intents of trip to overcome the cold-start problem and improve the prediction accuracy. At last, diversity is taken into consideration for cold start users. Our experiments show that the hybrid framework outperforms other pure latent factor models. In the meantime, the intent serves as an important feature in predicting users' missing rating.

Possible piece of future work is to modify the preference factor model by improving the feature extraction technique, and to find a better way for CF and CBF confusion.

ACKNOWLEDGEMENT

This work was supported by NSFC grant (No.61170085, 61472141 and 61321064), Program for New Century Excellent Talents in China (No.NCET-10-0388) and Shanghai Knowledge Service Platform Project （No. ZF1213）.

REFERENCES

[1] Andrew I. Schein, et al. "Methods and metrics for cold-start recommendations." Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2002.

[2] Marina Drosou, and Evaggelia Pitoura. "Disc diversity: result diversification based on dissimilarity and coverage." Proceedings of the VLDB Endowment 6.1 (2012): 13-24.

[3] Xiaoyuan Su, and Taghi M. Khoshgoftaar. "A survey of collaborative filtering techniques." Advances in artificial intelligence 2009 (2009): 4.

[4] Arkadiusz Paterek. "Improving regularized singular value decomposition for collaborative filtering." Proceedings of KDD cup and workshop. Vol. 2007. 2007.

[5] Daniel D. Lee, and H. Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization." Nature 401.6755 (1999): 788-791.

[6] Andriy Mnih, and Ruslan Salakhutdinov. "Probabilistic matrix factorization." Advances in neural information processing systems. 2007.

[7] Ruslan Salakhutdinov, and Andriy Mnih. "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo." Proceedings of the 25th international conference on Machine learning. ACM, 2008.

[8] Minqing Hu, and Bing Liu. "Mining and summarizing customer reviews." Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2004.

[9] Julian McAuley, and Jure Leskovec. "Hidden factors and hidden topics: understanding rating dimensions with review text." Proceedings of the 7th ACM conference on Recommender systems. ACM, 2013.

[10] Yang Bao, and Hui Fang Jie Zhang. "TopicMF: Simultaneously Exploiting Ratings and Reviews for Recommendation." (2014).

[11] Yongfeng Zhang, et al. "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis." Proceedings of SIGIR. Vol. 14. 2014.

[12] Jovian Lin, et al. "New and improved: modeling versions to improve app recommendation." Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval. ACM, 2014.

[13] Asher Levi, et al. "Finding a needle in a haystack of reviews: cold start context-based hotel recommender system." Proceedings of the sixth ACM conference on Recommender systems. ACM, 2012.

[14] Jaime Carbonell, and Jade Goldstein. "The use of MMR, diversity-based reranking for reordering documents and producing summaries." Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1998.