Cross-Domain Recommendation in the Hotel Sector

Marie Al-Ghossein
LTCI, Télécom ParisTech
Paris, France
malghossein@enst.fr

Talel Abdessalem
LTCI, Télécom ParisTech
Paris, France
UMI CNRS IPAL NUS
talel.abdessalem@enst.fr

Anthony Barré
AccorHotels
Paris, France
anthony.barre@live.fr

ABSTRACT

Hotel recommendation suffers from a severe sparsity problem. Travelers only book hotels once or twice a year, and one booking dataset may not gather all the bookings done by one user. Cross-domain recommendation can be leveraged to face the sparsity problem by exploiting knowledge from a related domain where feedback can be easily collected. In this paper, we propose to leverage check-ins data from location based social networks to learn mobility patterns and use it for hotel recommendation, considering that the choice of destination is an important factor for hotel selection. We present our developed solution, where we map items and users from both domains based on a number of observations, learn preferences for regions and for hotels, and combine the results to perform the final recommendation. Experiments on a real booking dataset using a dataset of geolocated posts show the interest of using data from other domains to boost hotel recommendation.

1 INTRODUCTION

Recommendations in the travel and tourism domains have become essential with the exponential growth of available data on the Web which have turned trip planning into a tiring and time-consuming task [7]. In particular, hotel recommender systems (RS) help users in choosing an appropriate option for accommodation [2]. The high-stakes nature of selecting accommodations also leads to the necessity of guiding users when making a decision.

While RS have been deployed in many domains, hotel recommendation must take into account the constraints considered by users when choosing a hotel, which are not present in other domains. In addition, hotel recommendation suffers greatly from sparsity since traveling is not a frequent activity [4]. Users only travel a few times each year and the feedback collected is sometimes not enough to learn user preferences. Sparsity constitutes therefore a major limitation for collaborative filtering approaches.

One way to address the sparsity problem is to leverage knowledge from other related domains where it is easier to get information regarding the behavior of users. Cross-domain RS [8] take advantage of the abundance of heterogeneous data providing multiple views of users' preferences. They aim to improve recommendations in a target domain by exploiting preferences uncovered in source domains. When applied in the tourism domain, cross-domain RS can suggest, for example, hotels based on flight bookings or events to attend based on hotel bookings [3].

When organizing a trip, travelers usually select the destination to visit before choosing the hotel where they will stay and the choice of accommodation highly depends on its location. Choosing a destination to visit is in turn related to several factors. First, the majority of trips are meant to explore destinations which are close to the place of residence of travelers. Then, users tend to follow the actual trends running locally which are also likely to change with time. In addition, the timing of the trip has an impact on the chosen destination. Some destinations are more popular during summer than winter, and leisure trips are more frequent during vacation periods.

Since hotel bookings are collected by organizations managing a subset of hotels and accommodations, hotel booking datasets do not cover all trips done and destinations visited by users. On the other hand, recent years have witnessed the emergence of Location Based Social Networks (LBSN), e.g., Flickr and Foursquare, where the mobility of users is captured through their check-ins. When exploring points-of-interest, users share their experiences on LBSN, making them a rich data source to analyze travel experiences.

In this paper, we address the problem of hotel recommendation suffering from sparsity by leveraging check-ins data from LBSN. We learn mobility patterns from the check-ins which are easily shared on LBSN and use them in combination with hotel preferences in order to boost hotel recommendation. We first map check-ins and hotels to a common space of geographical regions based on the density of hotels spread worldwide. We learn preferences for geographical regions based on check-ins data, link users from both domains, and combine the preferences in order to generate recommendations. Experiments on a dataset of hotel bookings extracted from the hotel industry show the interest of using LBSN data.

The rest of the paper is organized as follows. In Section 2, we discuss related work on hotel recommendation and cross-domain recommendation. In Section 3, we present our approach for hotel recommendation leveraging mobility data from LBSN. Experiments and results are discussed in Section 4. Finally, Section 5 concludes the paper.

2 RELATED WORK

Hotel recommendation. Several data sources have been exploited in previous work to address the problem of hotel recommendation. Saga et al. [19] rely on *implicit feedback* in the form of booking transactions to build a hotel-user graph which is used as a preference transition network. Other proposed approaches consider *explicit feedback* in the form of textual reviews. Along this line, Levi et al. [12] use reviews written by users with similar background. Similarity is measured based on a set of criteria including the nationality, the travel intention, and preferences for hotel traits. Zhang et al. [24] use textual reviews to model users and hotels in latent topic spaces generating hotel and user similarity matrices. Nilashi et al. [15] leverage ratings on several aspects of hotels including the location, the cleanliness, and the value, among others. Hotel recommendation can also benefit from *contextual dimensions*. When addressing the problem of lodging recommendation, Sanchez-Vazquez et al. [20] consider several dimensions like the price sensitivity, the perceived value, and the risk involved in the selection, among others.

Even though destination is an important parameter for hotel selection, the problem of hotel recommendation is different than the one of point-of-interest recommendation [23]. Hotel visits occur while on trips that are separated by a return to the user's place of residence, while points-of-interest are frequently visited sequentially. To the best of our knowledge, this is the first work exploiting cross-domain information for the benefit of hotel recommendation.

Cross-domain recommendation. Cross-domain RS [8] aim to improve recommendation in a target domain by leveraging user preferences from a source domain. The main advantages of using cross-domain recommendation include diversifying recommendations, addressing the cold-start problem, and alleviating the sparsity problem. It is therefore possible to suggest songs to listen to based on users' preferences for movies, for example.

Passing from one domain to the other requires considering the overlap between users and items or the similarities between item features and user behavior, in the different domains. Cremonesi et al. [6] defines four scenarios for cross-domain recommendation derived from the overlapping possibilities of users and items: There could be no overlap between users and items from both domains, overlap between users, overlap between items, or overlap between both users and items.

Several techniques have been developed to perform cross-domain recommendation. Cantador et al. [5] presents a categorization of these approaches and distinguishes between two classes. The first one relies on aggregating knowledge collected from the various domains in order to perform recommendation. One way to do this is by merging user preferences in the form of ratings for example [14] or combining the recommendations from the various domains [9]. The second class of techniques manages to transfer knowledge from one domain to the other. This is done through sharing latent features [16] or transferring rating patterns [13].

In this work, we use basic approaches for cross-domain recommendation to leverage data from LBSN and we show the interest of using them for hotel recommendation.

3 OUR PROPOSED APPROACH

Motivation. In order to cope with the sparsity problem faced in hotel recommendation, we propose to learn mobility patterns from check-ins shared on LBSN and combine it with hotel preferences in order to generate recommendations. In the source domain $\mathcal{S}$, we have active users on LBSN, $\mathcal{U}_\mathcal{S}$, who share their check-in activity. The items $\mathcal{I}_\mathcal{S}$ are the geolocated points visited. The target domain $\mathcal{T}$ is the hotel domain where the users $\mathcal{U}_\mathcal{T}$ are the one booking hotels and the items to recommend, $\mathcal{I}_\mathcal{T}$, are the hotels. In the problem we are considering, there is no overlap between users from both domains as we are not able to link users posting on LBSN and users booking hotels. However, a mapping can be done between check-ins $\mathcal{I}_\mathcal{S}$ and hotels $\mathcal{I}_\mathcal{T}$ based on the corresponding location, and similarities between users from both domains, $\mathcal{U}_\mathcal{S}$ and $\mathcal{U}_\mathcal{T}$, can be computed based on the visited locations.

Our work is motivated by a number of ideas. First, our approach is inspired from the real decision-making process of users when choosing a hotel: They first select a destination to visit, and then a hotel where they can stay. The source domain contains users' paths through their check-in activity. We try to use the knowledge from the source domain to learn accessible destinations for users based on their history. Accessibility usually relies on distance, cost, value, and other hidden variables.

We are therefore interested in the mobility patterns at a high scale. In our problem, preferences for regions are more relevant than preferences for specific points-of-interest. Based on data from LBSN, we can get the set of cities visited by one user, for example, and use this information for recommendation. Once the destination is selected, the hotel choice is more likely to depend on its features. On the other hand, hotels are not equally spread worldwide. When considering regions where we have a high density of hotels, it would be relevant to learn preferences for different subregions. Since all the neighborhoods in a specific city do not have the same characteristics, travelers may prefer one over the other.

We consider therefore a decomposition of the world map in several regions, where the region size depends on the corresponding hotel density. Items from both domains, $\mathcal{I}_\mathcal{S}$ and $\mathcal{I}_\mathcal{T}$, are mapped to these regions. Using the behavior of users on LBSN, we learn the preferences of users to the defined regions.

To benefit from these insights, and since there is no overlap between users from both domains, we associate users from the target domain, $\mathcal{U}_\mathcal{T}$, with the ones that are the most similar from the source domain, $\mathcal{U}_\mathcal{S}$, with respect to the visited regions. Preferences for geographical regions and hotels are finally combined to generate hotel recommendations.

In the following, we detail each part of our approach. In this Section, a recommendation method designates any latent factor model [11] that can be used for uncovering latent factors representing users and items. A score is then computed for each hotel, and the items that get the highest score are proposed for recommendation.

Mapping items from both domains. As mentioned before, we decompose the world map into several regions denoted as $\mathcal{I}_\mathcal{R}$ and map check-ins and hotels to these regions based on their location. The world map decomposition depends on the density of hotels in each area: Regions with high density of hotels should be further decomposed into subregions.

We rely on a hierarchical division of the space into rectangular spaces used in [1] and inspired by the clustering approach STING [22]. The first level of the hierarchy covers the whole region considered and corresponds to the whole map which constitutes one cell. Each cell at a level l is partitioned into 4 cells at the next level $l+1$, and the maximum number of levels is fixed.

We cluster hotels using a top-down approach based on the hierarchical structure of cells. The density of hotels in one cell is defined as the number of hotels located there divided by the area of the cell. For every level, starting with the first one, we compute the density of hotels in each cell. We then compare it to the density of the parent cell: If it is higher, we consider it as a cluster, otherwise, we move to the next level and repeat the process. This process is maintained until all the hotels are clustered or until we reach the maximum level.

Each cell containing a cluster of hotels is included in $\mathcal{I}_\mathcal{R}$. Each item from the sets $\mathcal{I}_\mathcal{S}$ and $\mathcal{I}_\mathcal{T}$ can be associated to a region from $\mathcal{I}_\mathcal{R}$. Using the feedback from the source domain (i.e., LBSN data) and $\mathcal{I}_\mathcal{R}$, we learn preferences for different regions.

Mapping users from both domains. Our aim is to use the preferences of users in $\mathcal{U}_\mathcal{S}$ to regions in $\mathcal{I}_\mathcal{R}$ to infer the preferences of users in $\mathcal{U}_\mathcal{T}$ to these regions. In order to do so, and for each user from $\mathcal{U}_\mathcal{T}$, we compute its neighbors (i.e., most similar users) contained in $\mathcal{U}_\mathcal{S}$ using a similarity measure. Z_u denotes the set of most similar users in $\mathcal{U}_\mathcal{S}$ for the user $u \in \mathcal{U}_\mathcal{T}$. The similarity measure handles user profiles from both domains defined as a binary vector which dimension is equal to the cardinality of $\mathcal{I}_\mathcal{R}$. If the user visited a check-in or a hotel located in a specific region, its value in the vector is 1, otherwise it is 0. We aggregate the region scores computed for each neighbor to get the scores for the target user.

Hotel recommendation. Performing hotel recommendation for a target user $u \in \mathcal{U}_\mathcal{S}$ requires computing hotels' scores, denoted by s_{ui}, for each hotel $i \in \mathcal{I}_\mathcal{T}$. Hotels are then ordered according to their scores and the k hotels having the highest scores are selected for recommendation.

The score computed is the combination of two scores: one from the source domain denoted by $s^{\mathcal{S}}_{ur}$, i.e., a score revealing the region preference for region $r \in \mathcal{I}_\mathcal{R}$, and the other from the target domain denoted by $s^{\mathcal{T}}_{ui}$, i.e., a score revealing the hotel preference for hotel $i \in \mathcal{I}_\mathcal{T}$.

In the source domain, we build a recommendation model modeling the preferences of users in $\mathcal{U}_\mathcal{S}$ to regions in $\mathcal{I}_\mathcal{R}$ and enabling the computation of scores of regions $r \in \mathcal{I}_\mathcal{R}$ for each user $z \in \mathcal{U}_\mathcal{S}$, i.e., $s^{\mathcal{S}}_{zr}$. In the target domain, we build a recommendation model modeling the preferences of users in $\mathcal{U}_\mathcal{T}$ to hotels in $\mathcal{I}_\mathcal{T}$ and enabling the computation of scores of hotels $i \in \mathcal{I}_\mathcal{T}$ for each user $u \in \mathcal{U}_\mathcal{T}$, i.e., $s^{\mathcal{T}}_{ui}$.

Final recommendations are performed for users from $\mathcal{U}_\mathcal{T}$. The score revealing the region preference for a user $u \in \mathcal{U}_\mathcal{T}$ is the aggregation of scores for the most similar users in $\mathcal{U}_\mathcal{S}$. The score revealing the hotel preference for a user in $\mathcal{U}_\mathcal{T}$, $s^{\mathcal{T}}_{ui}$, is directly computed using the built model. Both scores are combined and the final score for hotel i located in region r is given as follows, having a predefined weight parameter α:

$$s_{ui} = \alpha . s^{\mathcal{T}}_{ui} + (1-\alpha) . \frac{\sum_{z \in Z_u} s^{\mathcal{S}}_{zr}}{|Z_u|} \tag{1}$$

In this work, we use a matrix factorization method, Bayesian Personalized Ranking [18], to learn preferences and compute scores since it performs well on our dataset of bookings. Any other recommendation method could be used within the same approach.

4 EXPERIMENTS

In this Section, we present the experiments we conducted to prove the interest of our approach.

Datasets. We used one dataset from each domain in order to test our approach. The hotel booking dataset is extracted from the hotel industry and contains bookings done by users during the last 3 years. It consists of 7.8M users, 4.5k hotels, and 34M bookings. Users come from all the world and hotels are spread in more than 90 countries.

We use YFCC [21], a real-world dataset published recently. It contains media objects which have been uploaded to Flickr between 2004 and 2014. A subset of the posts are annotated with geographic coordinates and can be used as check-ins. We consider users that have visited more than 5 regions from the one we define. The dataset we use contains around 24M check-ins done by 32k users.

Experimental setup. We split the booking dataset into a training and a test set. We sort the bookings of each user in a chronological order and select the first 80% of bookings as the training set and the rest as the test set. We also select 20% of the users who have only done one booking and add them to the test set in order to evaluate the performance on new users. We use the data from the training set to train our recommendation method and evaluate its performance on the test set.

Evaluation metrics. We consider that we recommend k hotels to each user and we note which of these hotels were actually visited by the user. We use $recall@k$ and $NDCG@k$ for measuring the performance. $recall@k$ is defined as follows:

$$recall@k = \frac{number\ of\ hotels\ the\ user\ visited\ among\ the\ top\ k}{total\ number\ of\ hotels\ the\ user\ visited}. \tag{2}$$

The Normalized Discounted Cumulated Gain (NDCG) measures the ranking quality and $NDCG@k$ is the normalized $DCG@k$ which is computed as follows:

$$DCG@k = \sum_{i=1}^{k} \frac{2^{y_i} - 1}{\log_2(i+1)}, \tag{3}$$

where y_i is a binary variable for the i-th hotel of the recommendation list, that is equal to 1 if the corresponding hotel is visited by the user and 0 otherwise. The $recall@k$ and $NDCG@k$ for the entire system are the average $recall@k$ and $NDCG@k$ over all evaluated users respectively.

Parameters. We performed a grid search over the parameter space of the methods in order to find the parameters that give the best performance. We report the performance corresponding to the parameters that lead to the best results.

Compared methods. We include in our comparison traditional recommendation methods that are listed in the following:

- **MostPop** recommends the most popular hotels to the users.
- **CB** is a content-based method where hotels and users are represented in the space of hotels' features using vector space models and tf-idf weighting [17]. Hotel features cover the location, the brand, the segment category, and offered services such as Wi-Fi connection, parking, meeting facilities, and children playground.
- **Knnu** is a user-centered neighborhood-based method where we use the Jaccard similarity measure and set the number of neighbors to 2000.
- **MF** is a matrix factorization technique handling implicit feedback [10]. We set the number of latent factors $K = 100$, the regularization parameters to 0.001 and $a = 1.0, b = 0.01$.
- **BPR** [18] is a matrix factorization technique that relies on pairwise preferences to learn the latent model. We set the number of factors $K = 100$ and the regularization parameters to 0.0025.
- **CD** is the method we propose in this paper, leveraging data from LBSN.

Results. Figure 1 shows the performance of the methods we consider. The results are represented for each category of users, defined by the number of bookings present in the training set. By definition, the metrics we use decrease when the number of bookings increases.

MostPop is the only method able to recommend hotels to inactive users (i.e., users with zero bookings in the training set). The inferiority of *CB* shows that users do not attribute a great importance to all the hotels' features considered. Further investigations showed that the location of the hotel is one of the few factors that greatly affect the decision. *Knnu* performs well for users with few bookings while *BPR* outperforms the other methods when the number of bookings increases significantly.

The results obtained for *CD* show the interest of using data from LBSN to alleviate the sparsity problem. *CD* outperforms all the other methods when the number of bookings is less than or equal to 10 bookings. The interest of using cross-domain information decreases when the number of bookings increases: *BPR* outperforms *CD* when the number of bookings is greater than 30.

One explanation may be due to the fact that the behavior of users actively sharing content on LBSN is not fully representative of the behavior of all travelers. In particular, people having done more than 30 bookings are more likely to be businesspeople which behavior is not necessarily similar to users on LBSN. In addition, once enough feedback about hotels is collected, it may be sufficient to learn hotel preferences and generate good recommendations. The interest of using *CD* is highlighted in the cold-start setting where hotel bookings alone are not enough to infer preferences. We note that the majority of users in the booking dataset have done less than 10 bookings and therefore, *CD* improves the global performance.

Discussion. This is the first work proposing to apply cross-domain recommendation in the hotel sector using, in particular, abundantly available data from LBSN. While it can be a promising approach especially in a sparse data environment, it opens several interesting challenges.

First, not all users have their mobility behavior represented by active users posting on LBSN. These users will not directly benefit from the proposed approach. A finer analysis of users posting on LBSN, i.e., users in the source domain, and users booking hotels, i.e., users in the target domain, may help identifying relevant user segments which behavior can be similar in both domains in terms of mobility and probably underrepresented in one or in both domains. As a first basic approach, we tried addressing this issue by defining segments based on the number of bookings made and comparing recommendation performances in each one, considering that the number of bookings made may reveal a certain aspect of the user category. Other alternatives including more advanced techniques may be applied. One possibility to benefit from cross-domain recommendation may be then to learn local models by user category.

Second, further advances in this direction should consider evaluating the recommendation diversity in terms of proposed locations. It is important to generate diverse recommendations and avoid suggesting hotels located in one same area. This may occur when one region gathering several hotels is promoted for a particular user.

Transferring knowledge from LBSN to the hotel sector may go beyond the mobility aspect by also considering temporality, i.e., periods during which one region is visited by specific users, and context of visits for example by analyzing metadata associated to the posts. In addition, while we used a clustering component to map both domains, other approaches for integrating knowledge may be exploited. For example, we may be considering to rely on a multi-task approach and to train models in both domains simultaneously.

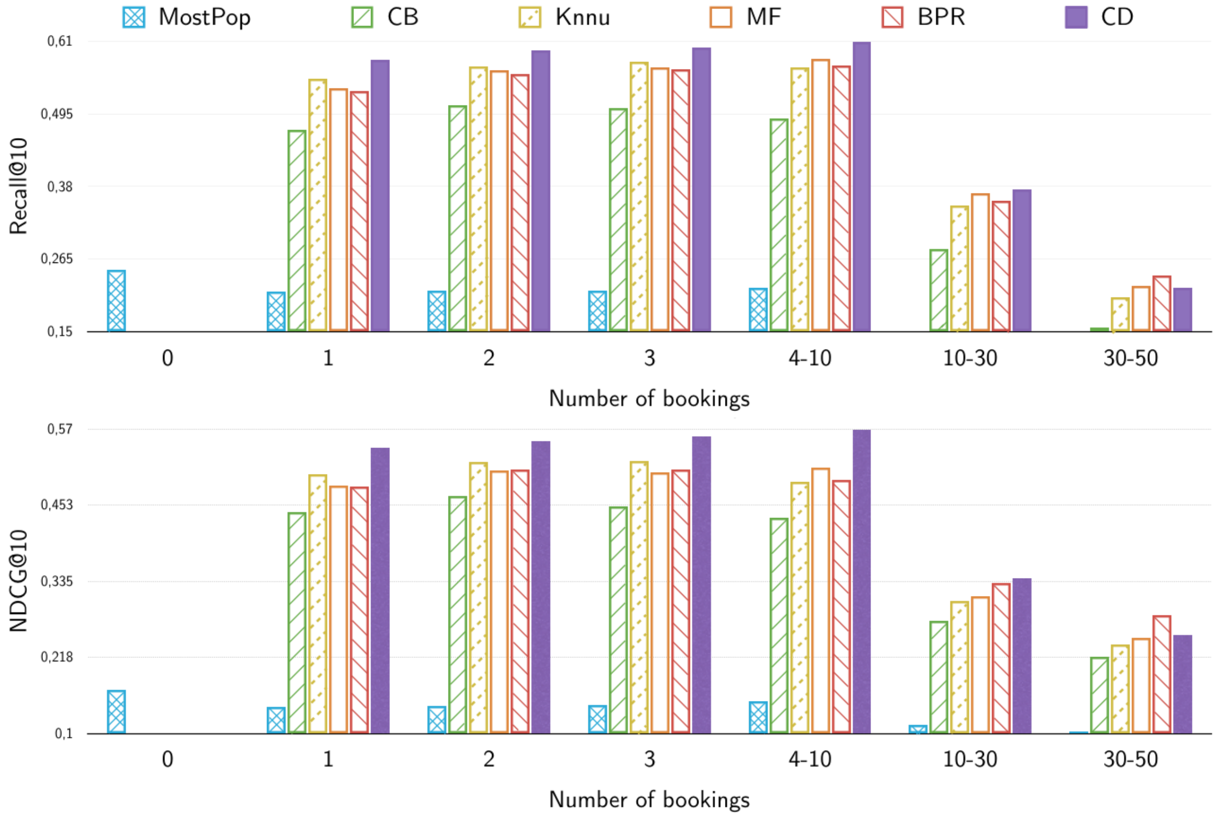

Figure 1: Recall@10 and NDCG@10 on the booking dataset. The results are represented based on the number of bookings in the training set.

5 CONCLUSION

In this paper, we propose to use data from LBSN to boost hotel recommendation. Hotel selection largely depends on the visited destination and some destinations are more accessible to users than others. Using the check-in activity from LBSN, we learn preferences for regions and use these preferences for hotel recommendation in order to address the sparsity problem. Mapping of items from both domains is done through a space of regions which definition is based on the density of hotels. Mapping of users from both domains is done by computing the similarity between users based on the visited locations. Hotel recommendation accounts for region preferences and hotel preferences. Experiments show the interest of using cross-domain information for users with few observations, i.e., in the cold-start setting.

Temporality plays an important role in the decision-making process: One destination is not considered by the same user in all periods of the year. Future work will involve adding the time dimension and taking into account in which period of the year the check-in was made in order to distinguish between users visiting the same destinations at different periods or seasons.

REFERENCES

[1] Marie Al-Ghossein and Talel Abdessalem. 2016. SoMap: Dynamic clustering and ranking of geotagged posts. In *Proceedings of the 25th International Conference Companion on World Wide Web*. International World Wide Web Conferences Steering Committee, 151–154.

[2] Marie Al-Ghossein, Talel Abdessalem, and Anthony Barré. 2018. Exploiting Contextual and External Data for Hotel Recommendation. In *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization (UMAP '18)*. 323–328.

[3] Marie Al-Ghossein, Talel Abdessalem, and Anthony Barré. 2018. Open data in the hotel industry: leveraging forthcoming events for hotel recommendation. *Information Technology & Tourism* (aug 2018). https://doi.org/10.1007/s40558-018-0119-6

[4] Lucas Bernardi, Jaap Kamps, Julia Kiseleva, and Melanie J. I. Müller. 2015. The Continuous Cold-start Problem in e-Commerce Recommender Systems. In *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems, Vienna, Austria, September 16-20, 2015*. 30–33.

[5] Iván Cantador and Paolo Cremonesi. 2014. Tutorial on Cross-domain Recommender Systems. In *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys '14)*. ACM, New York, NY, USA, 401–402. https://doi.org/10.1145/2645710.2645777

[6] Paolo Cremonesi, Antonio Tripodi, and Roberto Turrin. 2011. Cross-domain recommender systems. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*. Ieee, 496–503.

[7] Alexander Felfernig, Sergiu Gordea, Dietmar Jannach, Erich Teppan, and Markus Zanker. 2007. A short survey of recommendation technologies in travel and tourism. (2007), 1722.

[8] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. [n. d.]. Cross-domain recommender systems: A

survey of the state of the art. In *Spanish Conference on Information Retrieval. sn, 2012.*

[9] Sharon Givon and Victor Lavrenko. 2009. Predicting Social-tags for Cold Start Book Recommendations. In *Proceedings of the Third ACM Conference on Recommender Systems (RecSys '09)*. ACM, New York, NY, USA, 333–336. https://doi.org/10.1145/1639714.1639781

[10] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. Ieee, 263–272.

[11] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[12] Asher Levi, Osnat Mokryn, Christophe Diot, and Nina Taft. 2012. Finding a needle in a haystack of reviews: cold start context-based hotel recommender system. In *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 115–122.

[13] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction.. In *IJCAI*.

[14] Babak Loni, Yue Shi, Martha Larson, and Alan Hanjalic. 2014. Cross-domain collaborative filtering with factorization machines. In *European conference on information retrieval*. Springer, 656–661.

[15] Mehrbakhsh Nilashi, Othman bin Ibrahim, Norafida Ithnin, and Nor Haniza Sarmin. 2015. A multi-criteria collaborative filtering recommender system for the tourism domain using Expectation Maximization (EM) and PCA–ANFIS. *Electronic Commerce Research and Applications* 14, 6 (2015), 542–562.

[16] Weike Pan, Nathan N Liu, Evan W Xiang, and Qiang Yang. [n. d.]. Transfer learning to predict missing ratings via heterogeneous user feedbacks. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence, 2011.*

[17] Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The adaptive web*. Springer, 325–341.

[18] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

[19] Ryosuke Saga, Yoshihiro Hayashi, and Hiroshi Tsuji. 2008. Hotel recommender system based on user's preference transition. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*. IEEE, 2437–2442.

[20] Raul Sanchez-Vazquez, Jordan Silva, and Rodrygo L.T. Santos. 2017. Exploiting Socio-Economic Models for Lodging Recommendation in the Sharing Economy. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. 260–268.

[21] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2016. YFCC100M: The new data in multimedia research. *Commun. ACM* 59, 2 (2016), 64–73.

[22] Wei Wang, Jiong Yang, and Richard Muntz. 1997. STING : A Statistical Information Grid Approach to Spatial Data Mining. In *VLDB*.

[23] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 325–334.

[24] Kai Zhang, Keqiang Wang, Xiaoling Wang, Cheqing Jin, and Aoying Zhou. 2015. Hotel recommendation based on user preference analysis. In *Data Engineering Workshops (ICDEW), 2015 31st IEEE International Conference on*. IEEE, 134–138.