# Collaborative Filtering with User Ratings and Tags

Tengfei Bao[1], Yong Ge[2], Enhong Chen[1], Hui Xiong[2], Jilei Tian[3]

[1] University of Science and Technology of China
tfbao92@ustc.edu.cn, cheneh@ustc.edu.cn
[2] MSIS Department, Rutgers University
yongge@pegasus.rutgers.edu, hxiong@rutgers.edu
[3] Nokia Research Center, jilei.tian@nokia.com

## ABSTRACT

User ratings and tags are becoming largely available on Internet. While people usually exploit user ratings for developing recommender systems, the use of tag information in recommender systems remains under-explored. In particular, it is not clear how to use both user ratings and user tags in a complementary way to maximize the performances of recommender systems. To this end, we propose a novel collaborative filtering model based on probabilistic matrix factorization to predict users' interests to items by simultaneously utilizing both tag and rating information. Specifically, we first perform low-rank approximation for three matrices at the same time to learn the low-dimensional latent features of users, items and tags. Then, we predict one user's preference to an item as the product of the user and item latent features. Finally, experimental results on real-world data show that the proposed model can significantly outperform benchmark methods.

## 1. INTRODUCTION

Recommender systems [1] address the information overloaded problem by identifying user interests and providing personalized suggestions. As a major technique for recommendation, collaborative filtering aims at predicting the preference of a user by using available user ratings or taste information from many users. More formally, given $N$ users, $M$ items and a $M \times N$ preference matrix $R$, collaborative filtering is typically to predict the unknown ratings in R by using the available training ratings of $R$. Various collaborative filtering algorithms have been proposed in the literature to address this prediction problem. Generally, these algorithms can be categorized into two groups [1]: memory-based [2] and model-based methods [9, 5, 8].

Recently, in addition to user ratings, many recommender systems have enabled users to provide personalized tag information (e.g., keywords or phrases) to items. For example, CiteULike [3] allows users to put some keywords to annotate the reference. As a result, a lot of tags, which are associated with both users and items, have been collected on Internet. This tag information clearly reflects both users' interests and the topics of items. For example, if one user of a movie recommender system often annotates different items with tags as "Action" and "Comedy", probably this user has preference for action/comedy movies. Also, if a movie is usually labeled with tags as "Adventure" and "Sea", this movie is possibly related to sea adventure. In other words, in addition to the user-item interaction through user ratings, there are also user-tag interaction and item-tag interaction. Thus, it is possible to exploit such tag information to further identify users' interests in items and make better recommendation.

Indeed, some efforts have been made to exploit the tag information to improve recommender systems. For example, the researchers [4, 10] have utilized tag information to improve content-based (instead of collaborative filtering) recommender systems, while Zhen et al. [11] incorporated tag information into probabilistic matrix factorization (PMF) [9], i.e.,a collaborative filtering method. Basically, they assume similar users should have similar tagging history. Thus, a graph regularization based on user graph is formulated and added into the objective function of the PMF model as one more penalty item. However, in [11], tag information is still quite under-explored. First, tag information is only utilized as a side information incorporated into the PMF model. Also, they did not consider the user-tag interaction in the latent factor space. Finally, the item-tag interaction is not exploited in their method.

To that end, in this paper, we propose a probabilistic model, named TriFac, based on the PMF model to explore both tag information and rating information in an integrated way. Specifically, we represent tags, users and items in the same latent feature space. The pairwise interactions are modeled as the product of the pair of latent features. For example, individual item-user interaction (i.e., a rating) is given as the product of the user feature vector and the item feature vector. Along this line, we learn the low-dimensional latent features of tags, users and items by simultaneously performing the low-rank approximations for three observed matrices. Also, to avoid the overfitting problem, Gaussian priors are used on tag, user, item feature vectors, which essentially lead to $\ell_2$-regularization items in the objective functions of matrix low-rank approximation. A practical and efficient algorithm is also introduced to solve the derived objective function of the TriFac model.

Finally, experiments have been conducted on real-world data sets. The experimental results demonstrate that the TriFac model can effectively explore both tag and rating in-


*ContextDD'12,* August 12, 2012, Beijing, China.

formation in a complementary way, and TriFac significantly outperforms some well-known methods, such as PMF [9] and TagiCoFi [11], which incorporates tags as a side information.

## 2. PRELIMINARIES

In this section, we first provide some basic notations that will be used in this paper. Then we introduce the basic idea of latent factor model by briefly reviewing the well-known PMF model [9].

Suppose we have $N$ users, $M$ items and $K$ tags. Given the observed ratings and tags, we use a $M \times N$ matrix $R$ to represent the ratings. $R_{ji}$ is the rating value by user $i(i = 1, \cdots N)$ for item $j(j = 1, \cdots M)$. And we use a $K \times N$ matrix $P$ to represent to the interaction between user and tag. $P_{ki}$ is the frequency of tag $k(k = 1, \cdots K)$ used by user $i(i = 1, \cdots N)$. Also, we use a $K \times M$ matrix $Q$ to represent to the interaction between item and tag. $Q_{kj}$ is the frequency of tag $k(k = 1, \cdots K)$ applied to item $j(j = 1, \cdots M)$. To represent users, items and tags in the latent feature space, let $U \in R^{D\times N}$, $V \in R^{D\times M}$ and $W \in R^{D\times K}$ be user, item and tag latent feature matrices respectively. Furthermore, we use column vectors $U_i(i = 1, \cdots N)$, $V_j(j = 1, \cdots M)$ and $W_k(k = 1, \cdots K)$ to represent $D$-dimensional user-specific, item-specific and tag-specific latent feature vectors respectively.

In PMF model [9] as shown in Fig 1, the rating $R_{ji}$ is modeled as a linear dot product of $U_i^T$ and $V_j$, with a noise term that is zero mean Gaussian with constant variance: $\epsilon \sim \mathcal{N}(0, \sigma_R^2)$. Given the training data, i.e., observed ratings, the conditional likelihood over the observed ratings is defined as:

$$p(R|U, V, \sigma_R^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}[\mathcal{N}(R_{ji}|U_i^T V_j, \sigma_R^2)]^{I_{ji}}, \qquad (1)$$

where $\mathcal{N}(x|\mu, \sigma_R^2)$ denotes the probability density function of the Gaussian distribution with mean $\mu$ and variance $\sigma_R^2$, and $I_{ji}$ is the indicator variable that is equal to 1 if user $i$ rated item $j$ and equal to 0 otherwise. As we can see from Equation 1, elements as 0 in $R$ are treated as missing values. Then, zero-mean spherical Gaussian priors are also placed on user and item feature vectors:

$$p(U|\sigma_U^2) = \prod_{i=1}^{N}\mathcal{N}(U_i|0, \sigma_U^2\mathbf{I}), \quad p(V|\sigma_V^2) = \prod_{j=1}^{M}\mathcal{N}(V_j|0, \sigma_V^2\mathbf{I}).$$

To train this model with available ratings, we need to maximize the log of the posterior distribution over $U$ and $V$. Furthermore, given the fixed $\sigma_R^2$, $\sigma_U^2$ and $\sigma_V^2$, maximizing the log-posterior is shown to be equal to minimizing the sum-of-squared-errors objective function with quadratic regularization terms[9]:

$$\begin{aligned} E =& \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ji}(R_{ji} - U_i^T V_j)^2 + \\ & \frac{\lambda_U}{2}\sum_{i=1}^{N}||U_i||_F^2 + \frac{\lambda_V}{2}\sum_{j=1}^{M}||V_j||_F^2, \end{aligned} \qquad (2)$$

where $\lambda_U = \sigma_R^2/\sigma_U^2$, $\lambda_V = \sigma_R^2/\sigma_V^2$, and $||\cdot||_F$ denotes the Frobenius norm.

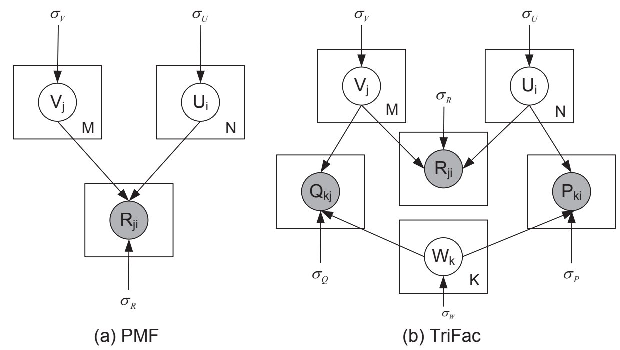


**Figure 1: Graphical Models.**

## 3. RECOMMENDATION WITH RATING AND TAG

In this section, we first introduce our TriFac model, which is able to model both rating and tag information in parallel. Then an effective algorithm is provided to learn the parameters of our model.

### 3.1 TriFac Model

In our TriFac model, user, item and tag are represented in the same latent feature space, as shown in Fig 1. Given the observed user-tag interaction, i.e., matrix $P$, we assume that high value of $P_{ki}$ indicates the high correspondence between user and tag latent features. More formally, we utilize inner product of $U_i^T$ and $W_k$ to model the interaction between user $i$ and tag $k$. Thus, the frequency of tag $k$ used by user $i$, i.e., $P_{ki}$, is approximated as: $\hat{P}_{ki} = U_i^T W_k$. Then, similar to PMF model, Gaussian noise with zero mean is assumed to the observed user-tag interaction matrix $P$. Thus, the conditional likelihood over $P$ is derived as:

$$p(P|U, W, \sigma_P^2) = \prod_{i=1}^{N}\prod_{k=1}^{K}\mathcal{N}(P_{ki}|U_i^T W_k, \sigma_P^2). \qquad (3)$$

$\mathcal{N}(x|\mu, \sigma_P^2)$ denotes the probability density function of the Gaussian distribution with mean $\mu$ and variance $\sigma_P^2$. Note that there are a lot of elements as 0 in matrix $P$, which indicates no interaction between users and tags. However, different from matrix $R$, these 0 values are more likely to indicate low correspondence between users and tags. In other words, these 0 values in $P$, not like those in $R$, do provide some helpful information. Actually, [6] has shown the evidences for this point. Thus, we still model the interactions and estimate the values of 0 in $P$ as inner product of tag and user latent features.

Similarly, for the observed matrix $Q$, we can gain the conditional likelihood over it as:

$$p(Q|V, W, \sigma_Q^2) = \prod_{j=1}^{M}\prod_{k=1}^{K}\mathcal{N}(Q_{kj}|V_j^T W_k, \sigma_Q^2), \qquad (4)$$

where we also use inner product of $V_j^T$ and $W_k$ to model the interaction between item $j$ and tag $k$, and place zero mean Gaussian noise. Moreover, we assume zero-mean spherical

Gaussian priors onto tag feature vectors as:

$$p(W|\sigma_W^2) = \prod_{k=1}^{K} \mathcal{N}(W_k|0, \sigma_W^2 \mathbf{I}). \quad (5)$$

Given the described linear modeling for interactions among each pair of user, item and tag, we can simultaneously utilize both tag and rating information. And the learning process can be done by performing low-rank approximation for the observed three matrices: $R$, $P$ and $Q$. Then the user, item and tag can be represented within the same latent feature space. Specifically, based on Fig 1 (b), we can derive the posterior distribution over user, item and tag feature as:

$$\begin{aligned} &p(U, V, W|R, P, Q, \sigma_R^2, \sigma_P^2, \sigma_Q^2, \sigma_U^2, \sigma_V^2, \sigma_W^2) \\ &\quad \propto p(R|U, V, \sigma_R^2)p(P|U, W, \sigma_P^2)p(Q|V, W, \sigma_Q^2) \\ &\qquad p(U|\sigma_U^2)p(V|\sigma_V^2)p(W|\sigma_W^2). \end{aligned} \quad (6)$$

Furthermore, the log of posterior distribution is given by:

$$\begin{aligned} &\ln p(U, V, W|R, P, Q, \sigma_R^2, \sigma_P^2, \sigma_Q^2, \sigma_U^2, \sigma_V^2, \sigma_W^2) \\ &\propto \ln p(R|U, V, \sigma_R^2) + \ln p(P|W, U, \sigma_P^2) + \ln p(Q|W, V, \sigma_Q^2) \\ &\quad + \ln p(U|\sigma_U^2) + \ln p(V|\sigma_V^2) + \ln p(W|\sigma_W^2) \\ &= -\frac{1}{2\sigma_R^2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ji}(R_{ji} - U_i^T V_j) \\ &\quad - \frac{1}{2\sigma_P^2}\sum_{i=1}^{N}\sum_{k=1}^{K}(P_{ki} - U_i^T W_k) \\ &\quad - \frac{1}{2\sigma_Q^2}\sum_{j=1}^{M}\sum_{k=1}^{K}(Q_{kj} - V_j^T W_k) \\ &\quad - \frac{1}{2\sigma_U^2}\sum_{i=1}^{N} {U_i}^T U_i - \frac{1}{2\sigma_V^2}\sum_{j=1}^{M} {V_j}^T V_j - \frac{1}{2\sigma_W^2}\sum_{k=1}^{K} W_k^T W_k \\ &\quad - \frac{1}{2}\left(NK ln\sigma_P^2 + MK ln\sigma_Q^2 + \left(\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ji}\right) ln\sigma_R^2\right) \\ &\quad - \frac{1}{2}\left(ND ln\sigma_U^2 + MD ln\sigma_V^2 + KD ln\sigma_W^2\right) + C, \end{aligned} \quad (7)$$

where $C$ is a constant which does not depend on parameters. Then, maximizing this log-posterior over user, item and tag features with parameters (such as $\sigma_R$ and $\sigma_U$) kept fixed is equivalent to minimizing the following objective function with quadratic penalty terms:

$$\begin{aligned} E = &\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ji}(R_{ji} - {U_i}^T V_j) + \frac{\lambda_P}{2}\sum_{i=1}^{N}\sum_{k=1}^{K}(P_{ki} - U_i^T W_k) \\ &+ \frac{\lambda_Q}{2}\sum_{j=1}^{M}\sum_{k=1}^{K}(Q_{kj} - V_j^T W_k) + \frac{\lambda_U}{2}\sum_{i=1}^{N}||U_i||_F^2 \\ &+ \frac{\lambda_V}{2}\sum_{j=1}^{M}||V_j||_F^2 + \frac{\lambda_T}{2}\sum_{k=1}^{K}||W_k||_F^2, \end{aligned} \quad (8)$$

where $\lambda_U = \sigma_R^2/\sigma_U^2$, $\lambda_V = \sigma_R^2/\sigma_V^2$, $\lambda_T = \sigma_R^2/\sigma_W^2$, $\lambda_P = \sigma_R^2/\sigma_P^2$, and $\lambda_Q = \sigma_R^2/\sigma_Q^2$.

## 3.2 Parameter Estimation

Although the objective function 8 is convex in $U$ only, $V$ only or $W$ only, it is non-convex in $U$, $V$ and $W$ together. However, two canonical types of algorithms can be applied to search the local minimal, including ALS (Alternating Least Squares) and Gradient Descent [7]. Since matrices $P$ and $Q$ contain no missing values, the looping over each element of $P$ or $Q$ will be very time-consuming [7]. Moreover, ALS allows us to parallelize the algorithm in order to search the minimal efficiently. Thus Gradient Descent is not practical here. Consequently, we employ ALS to search the local minimal of objective function 8. Specifically, we alternatively solve the optimization problem via fixed two latent feature matrices and iteratively update $U$, $V$ and $W$ as:

$$\begin{aligned} U_i &= \left(\sum_{j=1}^{M} {V_j}^T V_j I_{ji} + \lambda_P W^T W + \lambda_U \mathbf{I}\right)^{-1} \\ &\quad \times \left(\sum_{j=1}^{M} V_j R_{ji} I_{ji} + \lambda_P \sum_{k=1}^{K} W_k^T P_{ki}\right) \\ V_j &= \left(\sum_{i=1}^{N} U_i^T U_i I_{ji} + \lambda_Q W^T W + \lambda_V \mathbf{I}\right)^{-1} \\ &\quad \times \left(\sum_{i=1}^{N} U_i R_{ji} I_{ji} + \lambda_Q \sum_{k=1}^{K} W_k^T Q_{kj}\right) \\ W_k^T &= \left(\lambda_P U^T U + \lambda_Q V^T V + \lambda_T \mathbf{I}\right)^{-1} \\ &\quad \times \left(\lambda_P \sum_{i=1}^{N} U_i P_{ki} + \lambda_Q \sum_{j=1}^{M} V_j Q_{kj}\right) \end{aligned}$$

# 4. EXPERIMENTAL RESULTS

In this section, we provide an empirical evaluation of the performances of TriFac model on two real-world data sets.

## 4.1 Data Sets

The first data set is MovieLens[1], which contains 10 million ratings by 71,567 users on 10,681 movies. Also there are 100,000 tags that users have tagged on items. We firstly removed those tags that were used by less than 20 users or annotated on less than 20 movies. Then, we filtered out those users or items that are not related to the remaining tags. The final data set contains 2,096 users, 4,446 items, and 209 tags. Some statistics are listed in Table 3. The second real-world data set is collected from an online recommender system Douban[2] via using public APIs[3]. Douban allows users to rate Movies, Books, or Music and annotate them with tags. Tags in the Douban website play an important role in recommedation task. We crawled 10,000 users' movie ratings and tags. Then, we proceed the similar filtering as described above for MovieLens data. Finally, we obtain 5,516 users, 31,811 movies and 1,171 tags. Some statistics are also listed in Table 3. We can see that Douban Movie data set contains more tags than MovieLens data set.

## 4.2 Evaluation Metric

We select the MAE [11] as the evaluation metric, which is computed on testing set as follows: $MAE = \frac{\sum_i \sum_j I_{ji}|R_{ji} - \hat{R}_{ji}|}{\sum_i \sum_j I_{ji}}$,

[1]http://www.grouplens.org/node/73
[2]http://www.douban.com/
[3]http://www.douban.com/service/apidoc/

Table 1: MAE Comparisons on Movielens Data set.

| Split | D = 5 | | | D = 10 | | | D = 20 | | |
|---|---|---|---|---|---|---|---|---|---|
| | PMF | TagiCoFi | TriFac | PMF | TagiCoFi | TriFac | PMF | TagiCoFi | TriFac |
| 20-80 | 0.7328 | 0.6766 | 0.6284 | 0.7404 | 0.6788 | 0.6296 | 0.7414 | 0.6767 | 0.6292 |
| 40-60 | 0.6548 | 0.6370 | 0.6100 | 0.6560 | 0.6387 | 0.6112 | 0.6705 | 0.6392 | 0.6105 |
| 60-80 | 0.6231 | 0.6180 | 0.5984 | 0.6243 | 0.6158 | 0.5961 | 0.6261 | 0.6161 | 0.5949 |
| 80-20 | 0.6076 | 0.6055 | 0.5904 | 0.6021 | 0.6005 | 0.5821 | 0.6102 | 0.5980 | 0.5828 |

Table 2: MAE Comparisons on Douban Movie Data set.

| Split | D = 5 | | | D = 10 | | | D = 20 | | |
|---|---|---|---|---|---|---|---|---|---|
| | PMF | TagiCoFi | TriFac | PMF | TagiCoFi | TriFac | PMF | TagiCoFi | TriFac |
| 20-80 | 0.6461 | 0.6356 | 0.6031 | 0.6914 | 0.6443 | 0.6015 | 0.7427 | 0.6472 | 0.6025 |
| 40-60 | 0.6281 | 0.6238 | 0.5855 | 0.6403 | 0.6281 | 0.5847 | 0.6829 | 0.6281 | 0.5894 |
| 60-80 | 0.6108 | 0.5900 | 0.5784 | 0.6147 | 0.6143 | 0.5777 | 0.6505 | 0.6141 | 0.5793 |
| 80-20 | 0.6023 | 0.5811 | 0.5742 | 0.6058 | 0.5994 | 0.5752 | 0.6313 | 0.6052 | 0.5769 |

Table 3: Some Statistics

| Statistics | User | Item |
|---|---|---|
| MovieLens Data | | |
| Min. Number of Ratings | 16 | 1 |
| Max. Number of Ratings | 1,843 | 946 |
| Average Number of Ratings | 199.33 | 93.97 |
| Min. Number of Tags | 1 | 1 |
| Max. Number of Tags | 146 | 28 |
| Average Number of Tags | 4.26 | 2.93 |
| Douban Movie Data | | |
| Min. Number of Ratings | 1 | 1 |
| Max. Number of Ratings | 4,274 | 2,579 |
| Average Number of Ratings | 264.32 | 45.83 |
| Min. Number of Tags | 1 | 1 |
| Max. Number of Tags | 674 | 185 |
| Average Number of Tags | 45.63 | 6.81 |

where $R_{ji}$ is rating value of user i on item j, and $\hat{R}_{ji}$ is the corresponding prediction by the model.

## 4.3 Baseline Methods

We compare our TriFac model with two well-known recommendation methods. The first method is PMF [9] as described in Section 2. The second method is called TagiCoFi [11], which extends the PMF model by incorporating tag as side information.

## 4.4 Parameter Selection

As TriFac and two baseline methods are all based on Probabilistic Matrix Factorization, they share some common parameters, including $\lambda_U$ and $\lambda_V$. To make the comparison fair, we borrow the same parameters of TagiCoFi [11], i.e., setting the $\lambda_U$ and $\lambda_V$ as 1. Also we remain the same values for other parameters of TagiCoFi. For our TriFac model, we similarly set penalty parameter $\lambda_T$ as 1. In addition, we empirically specify the parameters $\lambda_P$ and $\lambda_Q$ of TriFac as 5 and 20 respectively. Specifically, we randomly split the Douban data set into 50% as training set and the other 50% as testing set. Then we compute MAE on testing set with 400 groups of $\lambda_P$ and $\lambda_Q$. Finally we select one group of $\lambda_P$ and $\lambda_Q$, which leads to the lowest MAE.

## 4.5 Performance Comparisons

In our experimental results, we split the data set into training part and testing part with different ratios, as shown in Tables 1 and 2. For example, 20-80 means the rating data is randomly splitted into 20% as training set and 80% as testing set. Also we compare TriFac model with PMF and TagiCoFi with different values of D, which is the number of latent factors. As we can see from Table 1, TriFac consistently outperforms both PMF and TagiCoFi with a significant margin on MovieLens data set. For Douban movie data set, the similar improvement by TriFac can be observed from Tabel 2.

In addition, we observe that the improvement of TriFac on Douban data set is more significant than that on MovieLens data set. To quantify this, we compute the average improved percentage of TriFac over PMF or TagiCoFi on each data set. Consequently, TriFac results in 10.21% improvement over PMF and 5.45% improvement over TagiCoFi on Douban data set, while TriFac leads to 8.61% improvement over PMF and 4.64% improvement over TagiCoFi on MovieLens data set. The reason is probably that there is more tag information for Douban data than MovieLens data. And our TriFac model effectively utilizes more tag information to enhance the prediction of rating.

# 5. CONCLUSION

In this paper, we proposed a novel collaborative filtering model, called TriFac, based on probabilistic matrix factorization, which is able to explore both rating and tag information in an integrated way. Specifically, we performed low-rank approximation for three matrices: item-user matrix, tag-user matrix and tag-item matrix, at the same time. Through the tri-matrices factorization, we can learn the user, item and tag features in the same latent space. The users's interest to items is then predicted as the inner product of user and item latent features. Finally, experimental results on real-world data sets illustrated the effectiveness of the TriFac model by comparing with two well-known baseline methods, such as PMF and TogiCoFi.

## 6. ACKNOWLEDGEMENTS

The work described in this paper was supported by grants from Natural Science Foundation of China (Grant No. 61073110), The National Major Special Science and Technology Projects (Grant No. 2011ZX04016-071), Research Fund for the Doctoral Program of Higher Education of China (20113402110024), and the Key Program of National Natural Science Foundation of China (Grant No. 60933013).

## 7. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Towards the next generation of recommender systems: A survey of the state-of-the art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[2] R. M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *IEEE ICDM*, 2007.

[3] CiteUlike. http://www.citeulike.org.

[4] M. de Gemmis, P. Lops, G. Semeraro, and P. Basile. Integrating tags in a semantic content-based recommender. In *RecSys*, 2008.

[5] Y. Ge, Q. Liu, H. Xiong, A. Tuzhilin, and J. Chen. Cost-aware travel tour recommendation. In *SIGKDD*, San Diego, CA, 2011.

[6] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *IEEE ICDM*, 2008.

[7] R. Koren, Y. Bell and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[8] Q. Liu, E. Chen, H. Xiong, and C. H. Q. Ding. Exploiting user interests for collaborative filtering: interests expansion via personalized ranking. In *ACM CIKM*, pages 1697–1700, Toronto, Canada, 2010.

[9] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, 2008.

[10] S. Sen, J. Vig, and J. Riedl. Tagommenders: connecting users to items through tags. In *WWW*, 2009.

[11] Y. Zhen, W.-J. Li, and D.-Y. Yeung. Tagicofi: Tag informed collaborative filtering. In *ACM RecSys 2009*.