Cost-aware Travel Tour Recommendation

Yong Ge[1], Qi Liu[1,2], Hui Xiong[1], Alexander Tuzhilin[3], Jian Chen[4]

[1] Rutgers Business School, Rutgers University
hxiong@rutgers.edu, yongge@pegasus.rutgers.edu
[2] University of Science and Technology of China, feiniaol@mail.ustc.edu.cn
[3] Leonard N. Stern School of Business, New York University, atuzhili@stern.nyu.edu
[4] Tsinghua University, jchen@mail.tsinghua.edu.cn

ABSTRACT

Advances in tourism economics have enabled us to collect massive amounts of travel tour data. If properly analyzed, this data can be a source of rich intelligence for providing real-time decision making and for the provision of travel tour recommendations. However, tour recommendation is quite different from traditional recommendations, because the tourist's choice is directly affected by the travel cost, which includes the financial cost and the time. To that end, in this paper, we provide a focused study of cost-aware tour recommendation. Along this line, we develop two cost-aware latent factor models to recommend travel packages by considering both the travel cost and the tourist's interests. Specifically, we first design a cPMF model, which models the tourist's cost with a 2-dimensional vector. Also, in this cPMF model, the tourist's interests and the travel cost are learnt by exploring travel tour data. Furthermore, in order to model the uncertainty in the travel cost, we further introduce a Gaussian prior into the cPMF model and develop the GcPMF model, where the Gaussian prior is used to express the uncertainty of the travel cost. Finally, experiments on real-world travel tour data show that the cost-aware recommendation models outperform state-of-the-art latent factor models with a significant margin. Also, the GcPMF model with the Gaussian prior can better capture the impact of the uncertainty of the travel cost, and thus performs better than the cPMF model.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*

General Terms

Algorithms, Experimentation

Keywords

Cost-aware Recommendation, Matrix Factorization

KDD'11, August 21–24, 2011, San Diego, California, USA.

1. INTRODUCTION

Recent years have witnessed a revolution in digital strategy for travel industry. As a result, massive amounts of travel data have been accumulated, and thus provide unparallel opportunities for people to understand user behaviors and generate useful knowledge, which in turn deliver intelligence for real-time decision making in various fields, including that of travel tour recommendation.

Recommender systems address the information overloaded problem by identifying user interests and providing personalized suggestions. In general, there are three ways to develop recommender systems[2]. The first one is content-based. It suggests items which are similar to those a given user has liked in the past. The second way is based on collaborative filtering[16, 20, 17, 5]. In other words, recommendations are made according to the tastes of other users that are similar to the target user. Finally, a third way is to combine the above and have a hybrid solution [6]. However, the development of recommender systems for travel tour recommendation is significantly different from developing recommender systems for traditional domains, since the tourist's choice is directly affected by the travel cost which includes both the financial cost and various other types of costs, such as time and opportunity costs.

Indeed, there are some unique characteristics of travel tour data, which distinguish the tour recommendation from the traditional recommendation, such as movie recommendation. First, the prices of travel packages can vary a lot. For example, by examining the real-world tour logs collected by a travel company, we can find that the prices of packages can range from $50 to $10000. Second, the time cost of packages also varies very much. For instance, while some travel packages take less than 3 day, other packages may take more than 10 days. In traditional recommender systems, the cost for taking a recommended item, such as a movie, is usually not a concern for the customers. However, the tourists usually have the financial and time constraints for selecting a travel package. In fact, Figure 1 shows the cost distributions of some tourists. In the figure, each point corresponds to one user. As can be seen, both the financial and time costs vary a lot among different tourists. Therefore, for the traditional recommendation models, which do not consider the cost of travel packages, it is difficult to provide the right tour recommendation for the right tourists. For example, traditional recommender systems might recommender a travel package to a tourist who cannot afford it because of the price.

To address this challenge, in this paper, we propose a cost-aware recommender system, which aims to mine the

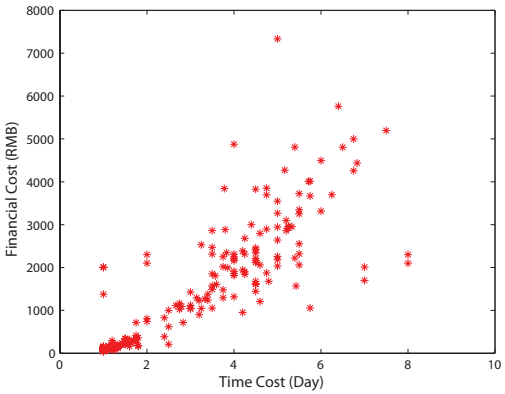

Figure 1: The Cost Distribution.

cost preferences and user interests simultaneously from the large scale of tour logs. The proposed recommender system is based on the latent factor model, which is one of the most popular approaches for collaborative filtering. In addition to the latent space features U_i and V_j that are widely used in latent factor models for recommender systems, such as the PMF model [24], we also introduce two types of costs into our model. The first type of costs refers to the observable costs of a tour package, defined in terms of the financial cost of the package and the time cost associated with the duration of the trip. For example, if a person goes on a trip to Cambodia for 7 days and pays $2000 for travel package j, then the observed package costs are denoted as a vector $C_{V_j} = (2000, 7)$. The second type of cost refers to unobserved costs associated with the user, which is denoted as a 2-dimensional latent cost vector C_{U_i}. Here, C_{U_i} is introduced to model user i's preference to the two aspects of cost.

The response of an active user to one specific item (tour package) is modeled as: $S(C_{U_i}, C_{V_j}) \cdot U_i^T \cdot V_j$, where S is the similarity function to measure the match of user and item cost. We denote this method as the cPMF model. In addition to the unknown latent features, such as U_i and V_j, user *cost vector* C_{U_i} also needs to be learned under the given loss function.

Furthermore, since there is still some uncertainty about the expense and the time cost that a user can afford, we further introduce a Gaussian priori $\mathcal{G}(C_{U_i})$, instead of using the *cost vector* C_{U_i}, on user cost to express the uncertainty. This Gaussian priori leads to an enhanced prediction model, denoted as the GcPMF model, which is trained by maximizing the posterior over the latent features and the parameters of Gaussian prior with observed data. In addition, efficient algorithms are developed to solve the optimization problems.

Finally, experiments on real-world tour logs show that both cPMF and GcPMF models outperform benchmark latent factor models with a significant margin. Moreover, the GcPMF model with the Gaussian priori leads to better performances than the cPMF model with the *cost vector*.

2. RELATED WORK

Two types of collaborative filtering models have been intensively studied recently (particularly with the incentive of NetFlix prize): memory-based and model-based approaches. Memory-based algorithms[10, 16, 5] essentially make rating prediction via using the rating of some other neighboring ratings. In the model-based approaches, training datasets are used to train a predefined model. Different approaches [14, 25, 29, 23]vary due to different statistical models assumed for the data. In particular, various matrix factorization (MF)[26, 24, 18, 3] methods have been proposed for collaborative filtering. Most MF approaches focus on fitting the user-item rating matrix using low rank approximation and use the learned latent user/item features to make predictions for unknown rating. In [24], PMF model was proposed by introducing Gaussian noise to observed rating. Under the Gaussian assumption, maximizing the posterior probability over latent features is equal to minimizing the square error, which is the objective function of most MF methods. More recently, more complex methods are proposed to consider user/item side information[1, 12], social influence[22], temporal information[19] and spatio-temporal context[21]. However, most of the above methods were developed for recommendation of movie, article, book or webpage, for which expense and time cost are usually not essential to the recommendation results. Including the latent item/user features, our cPMF and GcPMF models also explicitly address the two types of cost for travel recommendation.

Travel-related recommendation has been studied before. For instance, in [13], one probabilistic topic model was proposed to mine two types of topics, i.e.,local topics (e.g.,lava, coastline) and global topics(e.g.,hotel, airport) from travelogue on the website. Travel recommendation is performed by recommending destination, which is similar to a given location or relevant to a given travel intention, to a user. [7] presents UbiquiTO tourist guide for intelligent content adaptation. UbiquiTO uses a rule-based approach to adapt the content of the provided recommendation. [30] also deploys content adaptation approach for presenting tourist-related information. Both content and presentation recommendations are tailored to particular mobile devices and network capabilities. They use content-based, rule based and Bayesian classification methods to provide tourism-related mobile recommendations. [4] presents a method to recommend various places of interest for tourists using physical, social and modal types of contextual information (including mobile location-based contextual information). The recommendation algorithm is based on the Factor Model that is extended to model the impact of the selected contextual conditions on the predicted rating.

Also some works [15, 9, 8, 11]related to profit/cost-based recommender systems have been done. For instance, [15] studies the impact of firm's profit incentives on the design of recommender systems and identifies the conditions under which a profit-maximizing recommender recommends the item with highest margins and those under which it recommends the most relevant item. It also explores the mismatch between consumer and firm incentives and determines the social costs associated with this mismatch. The paper [9] studies the question of how a vendor can directly incorporate profitability of items into the recommendation process so as to maximize the expected profit while still providing accurate recommendations. The proposed approach takes the output of a traditional recommender system and adjusts it according to item profitability.

However, most of these prior travel-related and cost-related work did not explicitly consider the expense and time cost for travel recommendation. Also our travel tour recommendation specifically consider travel package recommendation by using large amount of users' travel logs collected from a travel agent.

3. COST-AWARE LATENT FACTOR MODELS

In this section, we first introduce a *cost-aware* probabilistic matrix factorization model, named cPMF. Then, we further exploit Gaussian prior on user costs and develop an *enhanced* cost-aware probabilistic matrix factorization model, called GcPMF. For both cPMF and GcPMF models, we derive the objective functions and develop effective solutions for the optimization problems.

3.1 The cPMF Model

cPMF is a cost-aware probabilistic matrix factorization model which represents user/item costs with 2-dimensional vectors as shown in Figure 2 (b). Suppose we have N users and M packages. Let R_{ij} be the rating of user i for package j, U_i and V_j represent D-dimensional user-specific and package-specific latent feature vectors respectively (both U_i and V_j are column vectors in this paper). Also, let C_{U_i} and C_{V_j} represent 2-dimensional cost vectors for user U_i and package V_j respectively. In addition, C_U and C_V simply denote the sets of all the user costs and all the package costs respectively. The conditional distribution over the observed ratings $R \in \mathbb{R}^{N \times M}$ is:

$$p(R|U,V,C_U,C_V,\sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}[\mathcal{N}(R_{ij}|f(U_i,V_j,C_{U_i},C_{V_j}),\sigma^2)]^{I_{ij}} \tag{1}$$

where $\mathcal{N}(x|\mu,\sigma^2)$ is the probability density function of the Gaussian distribution with mean μ and variance σ^2, and I_{ij} is the indicator variable that is equal to 1 if user i rated item j and is equal to 0 otherwise. The function $f(x)$ is to approximate the rating for item j by user i. Considering the cost preference for tour recommendation, we define $f(x)$ as:

$$f(U_i,V_j,C_{U_i},C_{V_j}) = S(C_{U_i},C_{V_j}) \cdot U_i^T V_j \tag{2}$$

where $S(C_{U_i},C_{V_j})$ is a similarity function to measure the similarity between user-cost vector and item-cost vector. Several existing similarity/distance functions can be used here to perform this calculation, such as Pearson coefficient, the cosine similarity or Euclidean distance. C_V can be considered to be known in this paper because we can directly obtain the cost information for tour packages from the tour logs. C_U is the user cost vector which is going to be estimated. Moreover, we also exploit zero-mean spherical Gaussian prior[24] on user and item latent feature vectors:

$$\begin{aligned} p(U|\sigma_U^2) &= \prod_{i=1}^{N}\mathcal{N}(U_i|0,\sigma_U^2\mathbf{I}), \\ p(V|\sigma_V^2) &= \prod_{j=1}^{M}\mathcal{N}(V_j|0,\sigma_V^2\mathbf{I}) \end{aligned} \tag{3}$$

As shown in Figure 2, in addition to user and item latent factor features, we also need to learn user cost vector simultaneously. Thus, by a Bayesian inference, we have

$$\begin{aligned} &p(U,V,C_U|R,C_V,\sigma^2,\sigma_U^2,\sigma_V^2) \\ &\propto\ p(R|U,V,C_U,C_V,\sigma^2)p(U|\sigma_U^2)p(V|\sigma_V^2) \\ &=\ \prod_{i=1}^{N}\prod_{j=1}^{M}[\mathcal{N}(R_{ij}|f(U_i,V_j,C_{U_i},C_{V_j}),\sigma^2)]^{I_{ij}} \\ &\quad\times\prod_{i=1}^{M}\mathcal{N}(U_i|0,\sigma_U^2\mathbf{I}) \times \prod_{j=1}^{N}\mathcal{N}(V_j|0,\sigma_V^2\mathbf{I}) \end{aligned} \tag{4}$$

U,V and C_U can be learned by maximizing this posterior or log-posterior over user-cost vectors, user and item features with hyperparameters (i.e. the observation noise variance and prior variance) being fixed. By Equation (4) or Figure 2, we can find that cPMF is actually an enhanced general model of PMF by taking the cost into consideration. In other words, if we limit $S(C_{U_i},C_{V_j})$ as 1 for all pairs of user and item, cPMF will be a PMF model.

The log of the posterior distribution in Equation (4) is:

$$\begin{aligned} &\ln p(U,V,C_U|R,C_V,\sigma^2,\sigma_U^2,\sigma_V^2) = \\ &-\tfrac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}(R_{ij}-f(U_i,V_j,C_{U_i},C_{V_j}))^2 \\ &-\tfrac{1}{2\sigma_U^2}\sum_{i=1}^{N}U_i^TU_i - \frac{1}{2\sigma_V^2}\sum_{j=1}^{M}V_j^TV_j - \frac{1}{2}[(\sum_{i=1}^{N} \\ &\sum_{j=1}^{M}I_{ij})\ln\sigma^2 + ND\ln\sigma_U^2 + MD\ln\sigma_V^2] + C, \end{aligned} \tag{5}$$

where C is a constant that does not depend on the parameters. Maximizing the log-posterior over user-cost vectors, user and item features is equivalent to minimize the sum-of-squared-errors objective function with respect to U, V and $C_U = (C_{U_1}, C_{U_2}, \cdots, C_{U_N})$:

$$\begin{aligned} E = &\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}I_{ij}(R_{ij}-S(C_{U_i},C_{V_j})\cdot U_i^TV_j)^2 \\ &+\tfrac{\lambda_U}{2}\sum_{i=1}^{N}||U_i||_F^2 + \frac{\lambda_V}{2}\sum_{j=1}^{M}||V_j||_F^2, \end{aligned} \tag{6}$$

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

Since the dimension of cost vectors is small [27], we use the Euclidean distance for the similarity function as $S(C_{U_i},C_{V_j}) = 1 - ||C_{U_i} - C_{V_j}||^2$. Since two attributes of the cost vector have significantly different levels of scale, we utilize the Min-Max Normalization technique to preprocess all cost vectors of items. Then the value of attribute of cost vectors is scaled to fit in a specific range $[0, 1]$. Subsequently, the value of the learned user cost vector and the similarity value also locate in the similar range. A local minimum of the objective function given by Equation (6) can be obtained by performing gradient descent in U_i, V_j and C_{U_i} as:

$$\begin{aligned} \frac{\partial E}{\partial U_i} &= \sum_{j=1}^{M}I_{ij}\left(S\left(C_{U_i},C_{V_j}\right)\cdot U_i^TV_j - R_{ij}\right) \\ &\quad\cdot S\left(C_{U_i},C_{V_j}\right)V_j + \lambda_U U_i \\ \frac{\partial E}{\partial V_j} &= \sum_{i=1}^{N}I_{ij}\left(S\left(C_{U_i},C_{V_j}\right)\cdot U_i^TV_j - R_{ij}\right) \\ &\quad\cdot S\left(C_{U_i},C_{V_j}\right)U_i^T + \lambda_V V_j \\ \frac{\partial E}{\partial C_{U_i}} &= \sum_{j=1}^{M}I_{ij}\left(S\left(C_{U_i},C_{V_j}\right)U_i^TV_j - R_{ij}\right) \\ &\quad\cdot U_i^TV_jS'\left(C_{U_i},C_{V_j}\right), \end{aligned} \tag{7}$$

where $S'\left(C_{U_i},C_{V_j}\right)$ is the derivative with respect to C_{U_i}. From this training process based on gradient descent, C_U can be eventually learned to express the user's cost.

3.2 The GcPMF Model

In real-world, the user expectation on the financial cost and the time of travel packages usually varies around certain values. Also, as shown in Equation (6), overfitting can happen if we perform the optimization with respect to C_{U_i} $(i = 1 \cdots N)$. These two observations suggest that it might be better if we could use a distribution to mode the user cost instead of representing it as a fixed 2-dimension vector. Therefore, we propose to use 2-dimensional Gaussian distribution to model the user cost in the GcPMF model as:

$$p(C_{U_i}|\mu_{C_{U_i}}, \sigma^2_{C_U}) = \mathcal{N}(C_{U_i}|\mu_{C_{U_i}}, \sigma^2_{C_U}\mathbf{I}). \quad (8)$$

In Equation (8), $\mu_{C_{U_i}}$ is the mean of the Gaussian distribution for the cost of user U_i. Also, $\mu_{C_{U_i}}$ is a 2-dimensional column vector. $\sigma^2_{C_U}$ is assumed to be the same for all the users for simplicity.

In the GcPMF model, since we use a 2-dimensional Gaussian distribution, instead of a 2-dimensional vector, to represent the user cost, we need to change the function to measure the similarity/match between the user's cost and the package cost [28]. Considering each package's cost is represented by a constant vector and the user's cost is characterized via a distribution, we can naturally measure the similarity between the user's cost and the package's cost as:

$$S_G(C_{V_j}, \mathcal{G}(C_{U_i})) = \mathcal{N}(C_{V_j}|\mu_{C_{U_i}}, \sigma^2_{C_U}\mathbf{I}), \quad (9)$$

where we simply use $\mathcal{G}(C_{U_i})$ to represent the cost distribution of user U_i. Please note that for the GcPMF model, C_{U_i} represents the variable of the user cost distribution $\mathcal{G}(C_{U_i})$, instead of a user cost vector. Furthermore, the function to approximate the rating for item j by user i is defined as:

$$\begin{aligned} f_G(U_i, V_j, \mathcal{G}(C_{U_i}), C_{V_j}) &= S_G(C_{V_j}, \mathcal{G}(C_{U_i})) \cdot U_i^T V_j \\ &= \mathcal{N}(C_{V_j}|\mu_{C_{U_i}}, \sigma^2_{C_U}\mathbf{I}) \cdot U_i^T V_j \quad (10) \end{aligned}$$

With this user cost representation and the similarity function, a similar Bayesian inference as Equation (4) is:

$$\begin{aligned} & p(U, V, \mu_{C_U}|R, C_V, \sigma^2, \sigma^2_U, \sigma^2_V, \sigma^2_{C_U}) \\ \propto\ & p(R|U, V, \mu_{C_U}, C_V, \sigma^2, \sigma^2_{C_U}) p(C_V|\mu_{C_U}, \sigma^2_{C_U}) p(U|\sigma^2_U) p(V|\sigma^2_V) \\ =\ & \prod_{i=1}^{N}\prod_{j=1}^{M} \left(\mathcal{N}\left(R_{ij}|f_G\left(U_i, V_j, \mathcal{G}(C_{U_i}), C_{V_j}\right), \sigma^2\right)\right)^{I_{ij}} \\ & \times \prod_{i=1}^{N}\prod_{j=1}^{M} \mathcal{N}(C_{V_j}|\mu_{C_{U_i}}, \sigma^2_{C_U}\mathbf{I})^{I_{ij}} \\ & \times \prod_{i=1}^{N} \mathcal{N}(U_i|0, \sigma^2_U\mathbf{I}) \times \prod_{j=1}^{M} \mathcal{N}(V_j|0, \sigma^2_V\mathbf{I}), \quad (11) \end{aligned}$$

where $\mu_{C_U} = (\mu_{C_{U_1}}, \mu_{C_{U_2}}, \cdots, \mu_{C_{U_N}})$, which denotes the set of means of all users' cost distribution. $p(C_V|\mu_{C_U}, \sigma^2_{C_U})$ is the likelihood given the parameters of user cost distribution. Given the known ratings of each user, the cost of packages rated by this user can actually be treated as observations of each user's cost. This is why we represent the likelihood over C_V, i.e. the set of package's cost. Then we are able to derive the likelihood as $\prod_{i=1}^{N}\prod_{j=1}^{M} \mathcal{N}(C_{V_j}|\mu_{C_{U_i}}, \sigma^2_{C_U}\mathbf{I})I_{ij}$.

The log of the posterior distribution in Equation (11) can be derived as:

$$\begin{aligned} & \ln p(U, V, \mu_{C_U}|R, C_V, \sigma^2, \sigma^2_U, \sigma^2_V, \sigma^2_{C_U}) = \\ & -\tfrac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij} - f_G\left(U_i, V_j, \mathcal{G}(C_{U_i}), C_{V_j}\right))^2 \\ & -\tfrac{1}{2\sigma^2_{C_U}}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(C_{V_j} - \mu_{C_{U_i}})^T(C_{V_j} - \mu_{C_{U_i}}) - \\ & \tfrac{1}{2\sigma^2_U}\sum_{i=1}^{N} U_i^T U_i - \frac{1}{2\sigma^2_V}\sum_{j=1}^{M} V_j^T V_j - \frac{1}{2}[(\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij})\ln\sigma^2 \\ & +(\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij})\ln\sigma^2_{C_U} + ND\ln\sigma^2_U + MD\ln\sigma^2_V] + C, \quad (12) \end{aligned}$$

where C is also a constant. Maximizing this log-posterior over user-cost means, user and item features is equivalent to minimize the sum-of-squared-errors objective function with respect to U, V and $\mu_{C_U} = (\mu_{C_{U_1}}, \mu_{C_{U_2}}, \cdots, \mu_{C_{U_N}})$:

$$\begin{aligned} E = & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}\left(R_{ij} - \mathcal{N}(C_{V_j}|\mu_{C_{U_i}}, \sigma^2_{C_U}\mathbf{I}) \cdot U_i^T V_j\right)^2 \\ & + \tfrac{\lambda_U}{2}\sum_{i=1}^{N}||U_i||_F^2 + \frac{\lambda_V}{2}\sum_{j=1}^{M}||V_j||_F^2 \\ & + \tfrac{\lambda_{C_U}}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}||C_{V_j} - \mu_{C_{U_i}}||^2, \quad (13) \end{aligned}$$

where $\lambda_{C_U} = \sigma^2/\sigma^2_{C_U}$, $\lambda_U = \sigma^2/\sigma^2_U$, $\lambda_V = \sigma^2/\sigma^2_V$. As we can see from Equation (13), the Gaussian prior introduced on user's cost leads to one more regularization term to the objective function, thus easing the over-fitting. The GcPMF model is also the enhanced general model of PMF, since the objective function (13) reduces to that of PMF if $\sigma^2_{C_U}$ is limited to be infinite. A local minimum of the objective function given by Equation (13) can be identified by performing gradient descent in U_i, V_j and $\mu_{C_{U_i}}$ as:

$$\begin{aligned} \frac{\partial E}{\partial U_i} = & \sum_{j=1}^{M} I_{ij}\left(\mathcal{N}(C_{V_j}|\mu_{C_{U_i}}, \sigma^2_{C_U}\mathbf{I}) \cdot U_i^T V_j - R_{ij}\right) \\ & \cdot\mathcal{N}(C_{V_j}|\mu_{C_{U_i}}, \sigma^2_{C_U}\mathbf{I})V_j + \lambda_U U_i \\ \frac{\partial E}{\partial V_j} = & \sum_{i=1}^{N} I_{ij}\left(\mathcal{N}(C_{V_j}|\mu_{C_{U_i}}, \sigma^2_{C_U}\mathbf{I}) \cdot U_i^T V_j - R_{ij}\right) \\ & \cdot\mathcal{N}(C_{V_j}|\mu_{C_{U_i}}, \sigma^2_{C_U}\mathbf{I})U_i^T + \lambda_V V_j \\ \frac{\partial E}{\partial \mu_{C_{U_i}}} = & \sum_{j=1}^{M} I_{ij}\left(\mathcal{N}(C_{V_j}|\mu_{C_{U_i}}, \sigma^2_{C_U}\mathbf{I})U_i^T V_j - R_{ij}\right) \cdot \\ & U_i^T V_j \mathcal{N}'(C_{V_j}|\mu_{C_{U_i}}, \sigma^2_{C_U}\mathbf{I}) + \lambda_{C_U}\sum_{j=1}^{M} I_{ij}(\mu_{C_{U_i}} - C_{V_j}), \quad (14) \end{aligned}$$

where $\mathcal{N}'(C_{V_j}|\mu_{C_{U_i}}, \sigma^2_{C_U}\mathbf{I})$ is the derivative with respect to $\mu_{C_{U_i}}$. For the same reason, we also utilize the Min-Max Normalization to preprocess all the cost vectors of item before training the model.

In the experiments, instead of using Equation (2) and Equation (10), which may have predictions out of the valid rating range, the results of Equation (2) and Equation (10) are thus further passed through the logistic function $g(x) = 1/(1 + exp(-x))$, which bounds the range of predictions as $[0, 1]$. Also, we map the ratings $1, \cdots, K$ (K is the maximum rating value) to the interval $[0, 1]$ using the function

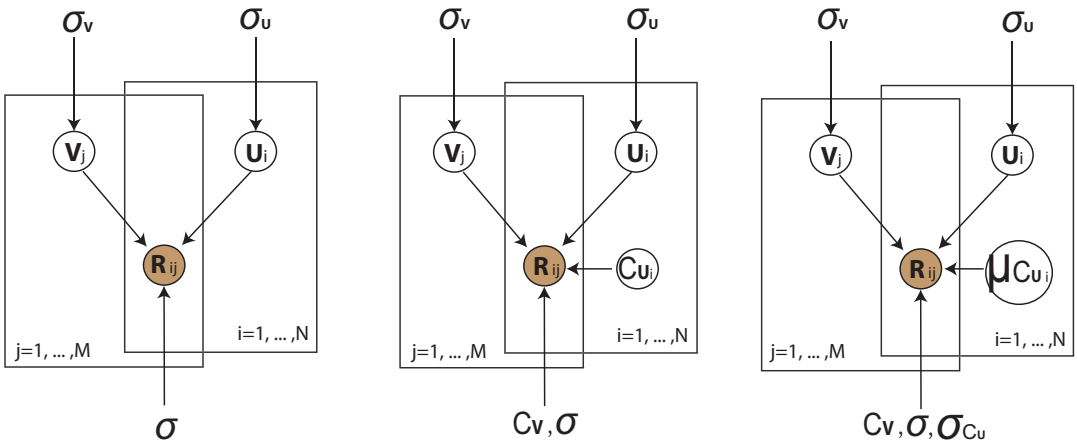

Figure 2: Graphical Models.

$t(x) = (x-1)/(K-1)$, thus the valid rating range matches the range of predictions by our models.

3.3 Discussion of the Model Efficiency

The main computation of gradient methods is to evaluate the object function and its gradients against variables. Because of the sparseness of matrices R, the computational complexity of evaluating the object function (6) is $\mathcal{O}(\eta f)$, where η is the number of nonzero entries in R and f is the number of latent factors. The computational complexity for gradients $\frac{\partial E}{\partial U}$, $\frac{\partial E}{\partial V}$ and $\frac{\partial E}{\partial C_U}$ in Equation (7) is also $\mathcal{O}(\eta f)$. Thus, for each iteration, the total computational complexity is $\mathcal{O}(\eta f)$. Thus, the computational cost of the cPMF model is linear with respect to the number of observed ratings in the sparse matrix R. Similarly, we can analyze and obtain the overall computational complexity of the GcPMF model is also $\mathcal{O}(\eta f)$ because the only difference between GcPMF and cPMF is that we need to compute the probability of the Gaussian distribution as the cost similarity instead of the Euclidean distance involved in cPMF. This complexity analysis shows that the proposed models are efficient and can scale to very large data. In addition, to speed-up training, instead of performing batch learning, we divide the training set into sub-batches and update the feature vectors and cost vectors/parameters after each sub-batch.

4. EXPERIMENTAL RESULTS

Here, we provide an empirical evaluation of the performances of cPMF and GcPMF on real-world travel tour data.

4.1 Travel Tour Data Description

The travel tour data set used in this paper is provided by a travel company. In the data set, there are more than 200,000 expense records starting from the beginning of 2000 to October 2010. In addition to the Customer ID and Package ID, there are many other attributes for each record, such as the start date, the travel days, the package name and some short descriptions of the package, and the cost of the package. Also, the data set includes some information about the customers, such as age and gender. From these records, we are able to obtain the information about users (tourists), items (packages) and user ratings. Instead of using explicit rating (i.e. scores from 1 to 5), we use the number of visits as the implicit rating. This is similar to the use of the number of clicks for measuring the user interests in Web pages. Moreover, we are able to know the financial and time cost for each package from these tour logs. Finally, the tourism data is much sparser than the movie data. For instance, a user can usually watch more than 12 movies each year, while there are not many people who will travel more than 12 times every year. In fact, many tourists only have one or two travel records in the data set.

To reduce the challenge of sparseness, we simply ignore users, who have traveled less than 4 times, as well as packages which have been used for less than 4 times. After this preprocessing, we have 34007 pairs of ratings with 1384 packages and 5724 users. However, the sparseness is still quite low, i.e. 0.4293%, which is much lower than the famous Movielens data set [1] with sparseness as 4.25% and Eachmovie [2] with sparseness as 2.29%. Actually, in the Movielens data set, all users have rated at least 20 movies, while in our travel data, 85.66% users have available ratings less than 10. For our travel log data, instead of using explicit rating, we use implicit rating; that is, the number of travels. Since a user may purchase the same package multiple times for her/his family members, and many local travel packages are even visited multiple times by the same user. There are still a lot of implicit ratings larger than 1, though over 50% of implicit ratings are 1. Some statistics of the item-user rating matrix are summarized in Table 1. From the total pairs of ratings, we randomly select 5% pairs of ratings as the test data for evaluation.

Table 1: Some Characteristics of Travel Data

Statistics	User	Package
Min Number of Rating	4	4
Max Number of Rating	62	1976
Average Number of Rating	5.94	24.57

4.2 Evaluation Metrics

The Root Mean Square Error (RMSE) is used to measure the prediction quality in comparison with benchmark collaborative filtering methods. The RMSE is defined as:

$$RMSE = \sqrt{\frac{\sum_{ij} (r_{ij} - \hat{r}_{ij})^2}{N}}, \qquad (15)$$

[1] http://www.cs.umn.edu/Research/GroupLens.
[2] HP retired the EachMovie dataset

where r_{ij} denotes the rating of item j by user i, $\hat{r}_{ij}$ denotes the corresponding rating predicted by the model, and N denotes the number of tested ratings.

In addition to RMSE, the Cumulative Distribution (CD) [16] is also applied for evaluating the performances of different models. Essentially, CD is designed to measure the qualify of top-K recommendations. The CD measurement could explicitly guide people to specify K in order to contain the most interesting items in the suggested top-K set with certain probability. In the following, we briefly introduce how to compute CD with the testing set(more details about this validation method can be found in [16]). First, all highest ratings in the testing set are selected. Assume that we have $\mathcal{M}$ ratings with the highest rating. For each item i with the highest rating by user u, we randomly select $\mathcal{C}$ additional items and predict the ratings by u for i and other $\mathcal{C}$ items. Then, we order these $\mathcal{C}+1$ items based on their predicted ratings in a decreasing order. There are $\mathcal{C}+1$ different possible ranks for item i, ranging from the best case where none(0%) of the random $\mathcal{C}$ items appearing before item i, to the worst case where all (100%) of the random $\mathcal{C}$ items appearing before item i. For each of those $\mathcal{M}$ ratings, we independently draw the $\mathcal{C}$ additional items, predict the associated ratings, and derive a relative ranking (RR) between 0% and 100%. Finally, we analyze the distribution of overall $\mathcal{M}$ RR observations, and estimate the cumulative distribution (CD). In our experiments, we specify $\mathcal{C} = 200$ and obtain 761 RR observations in total.

4.3 The Details of Training

Here, we introduce the training details for the comparison of SVD, PMF [24], cPMF, and GcPMF models. The SVD model was trained to minimize the sum-squared error only to the observed entries of the target matrix. The feature vectors of the SVD model were not regularized in any way. For the PMF model, we empirically specified the parameters as: $\lambda_U = 0.05$ and $\lambda_V = 0.005$. For cPMF and GcPMF models, we used the same values for λ_U and λ_V, together with $\lambda_{C_U} = 0.2$ for the GcPMF model. Also, we specified $\sigma^2_{C_U} = 0.09$ for the GcPMF model in the following. Moreover, we initialized the cost vector of each user or the Gaussian mean vector with the average cost of all items rated by this user u, while user/item latent feature vectors were initialized randomly. Finally, we simply removed the global effect [20] by subtracting the average rating of the training set from each rating before performing collaborative filtering. All the models were compared with the test set.

4.4 Performance Comparisons

First, a performance comparison of all the models with 10-dimensional latent features in terms of RMSE is shown in Figure 3. As can be seen, both cPMF and GcPMF outperform PMF or SVD with significant margins, since the RMSE values achieved by cPMF or GcPMF are much lower than those achieved by PMF or SVD. Also, the performances of GcPMF are consistently better than that of cPMF. In addition, given the same parameters and the initialization setting, both cPMF and GcPMF lead to faster convergence than PMF and SVD. Towards the end of training, SVD has the serious overfitting issues. The above results suggest that it is essential to consider the travel cost for travel tour recommendation and the proposed two cost-aware models can

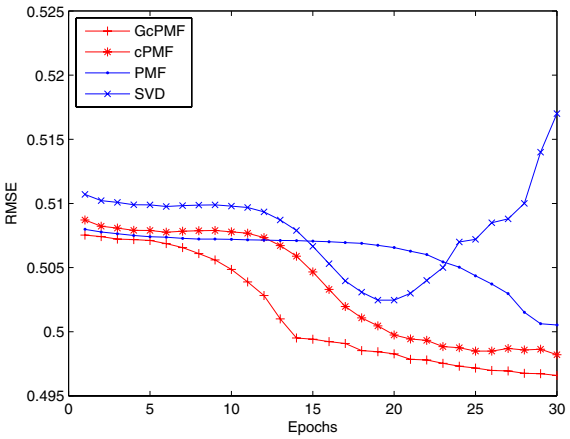

Figure 3: A Performance Comparison in terms of RMSE (10D Latent Features).

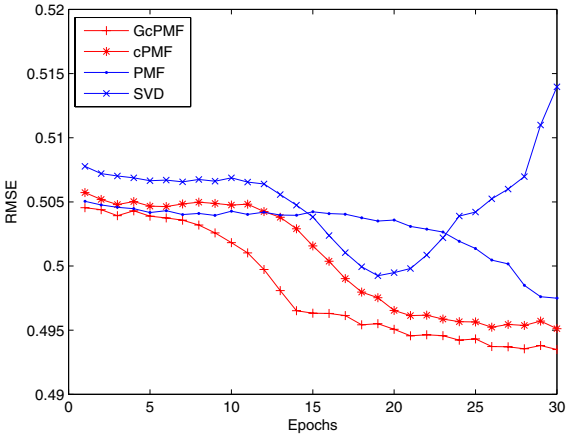

Figure 4: A Performance Comparison in terms of RMSE (30D Latent Features).

capture the cost effect well. Finally, in Figure 4, similar results can also be observed for 30-dimensional latent features.

Second, we compared the performances of all the models using the CD measure introduced in Subsection 4.2. Figure 5, shows the cumulative distribution of the computed percentile ranks for the four models over all 761 RR observations. Note that we used 10-dimensional latent features in Figure 5. As can be seen, both cPMF and GcPMF models outperform the competing models. For example, considering the point of 0.1 on x-axis, the CD value for GcPMF at this point suggests that, if we recommend top-20 from random 201 packages to users, approximately at least one package matches user interest and cost-expectation with probability as 53%. Since people usually are more interested in top-5 or even top-3, out of 201 packages, we zoom in on the head of the x-axis, which represent top-K recommendation in a more detailed way. As shown in Figure 6, a more clear difference can be observed. For example, GcPMF model has a probability of almost 0.5 to suggest a highest-rated package before other random 198 packages. In other words, if we use GcPMF to recommend top-2 packages out of 201 packages, we can match user's needs with probability of 0.5. This outperforms PMF and SVD with over 60% percentage and around 20% percentage respectively. Also, cPMF leads to

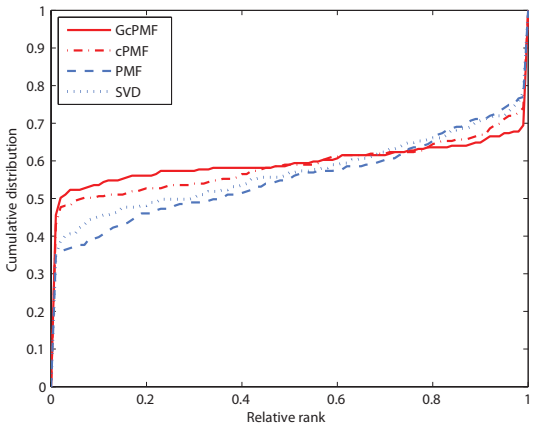

Figure 5: A Performance Comparison in terms of CD (10D Latent Features).

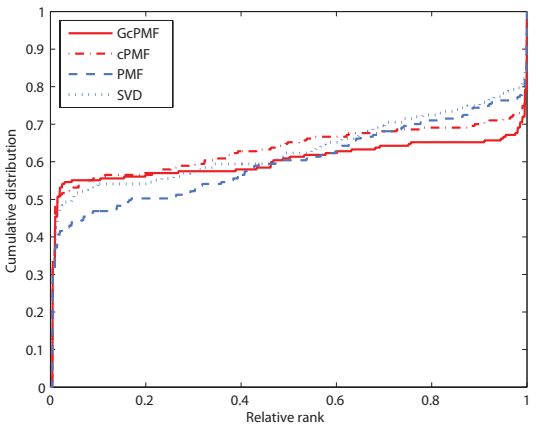

Figure 7: A Performance Comparison in terms of CD (30D Latent Features).

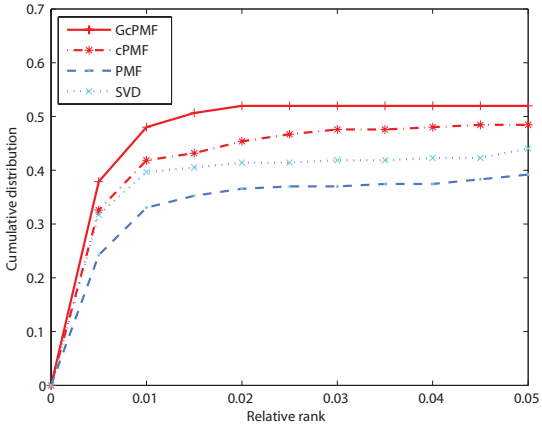

Figure 6: A Local Performance Comparison in terms of CD (10D Latent Features).

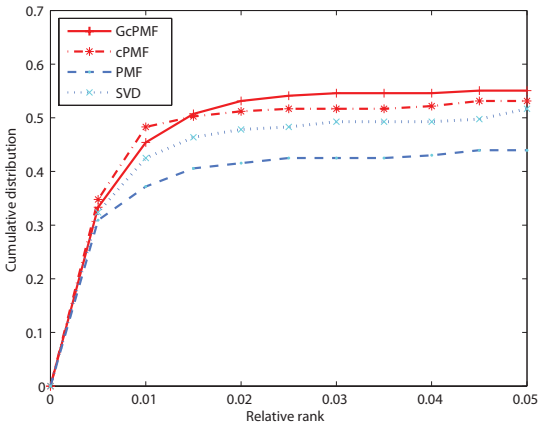

Figure 8: A Local Performance Comparison in terms of CD (30D Latent Features).

slight better performance than SVD and much better performance than PMF. In addition, we show more comparisons in Figures 7 and 8 with 30-dimensional latent features, where a similar trend can be observed.

4.5 Cost Visualization

In this subsection, we illustrate the user cost learned by our models. These learned user cost features could help travel companies for customer profiling. Since we normalized the package cost vectors into $[0, 1]$ before feeding into our models, the learned user cost features (C_U and μ_{C_U}) via our models have the similar scale as normalized package cost vectors. To visualize the learned C_U, we first restored the scale of user cost features (C_U and μ_{C_U}) by using the inverse transformation of MinMax normalization. Figure 9 shows the financial cost feature of C_U for randomly-selected 40 users, where each user corresponds to a column of vertically-distributed points. Neighboring users are differentiated with different colors. For example, for the right vertical blue points, *star* represents the learned user financial cost feature and *dot* represents the financial cost of packages, which are rated by this specific user in the training set. As we can see, the learned user financial cost feature is relatively representative. However, there is still obvious variance among the package cost features by some users. That is why we apply Gaussian distribution to model user cost. To illustrate the effectiveness of Gaussian assumption for user cost, in Figure 10, we visualize the learned μ_{C_U} for randomly-selected 12 users. For each subfigure of Figure 10, we directly plot the learned 2-dimension μ_{C_U} (without inverse transformation) for individual user and all normalized 2-dimension cost vectors of packages, which are rated by the user in the training set. And μ_{C_U} is represented as *star* and *dot* represents package cost vector.

4.6 Performances on Different Users

For the four competing models, the prediction performances for users with different number of observed ratings usually vary a lot, because the user feature vectors are updated differently during the training process. The user cost vector of cPMF or the Gaussian distribution of GcPMF play as an effective constraint to limit the prediction via the similarity weight. Thus, the performance of cPMF and GcPMF on users with few observed ratings are expected to be better than the competing models. Figure 11 shows this effect by comparing RMSE on different users for the four models. In this experiment, we used 10-dimensional latent factors, and the users were grouped by the number of observed ratings in the training set. Also, we used the RMSE value of final epoch for all models. As we can see, even for users with less than 6 observed ratings, cPMF and GcPMF lead to better performances in terms of RMSE.

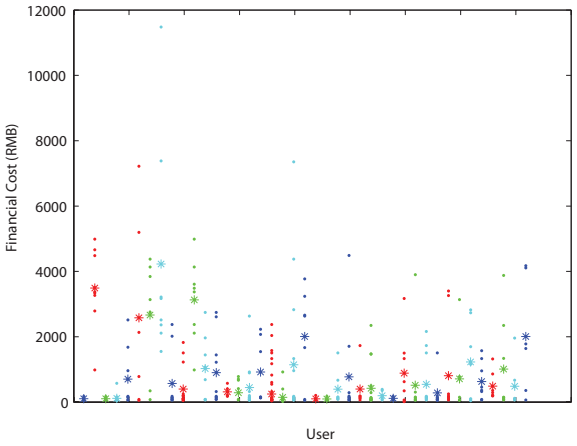

Figure 9: An Illustration of User Financial Cost.

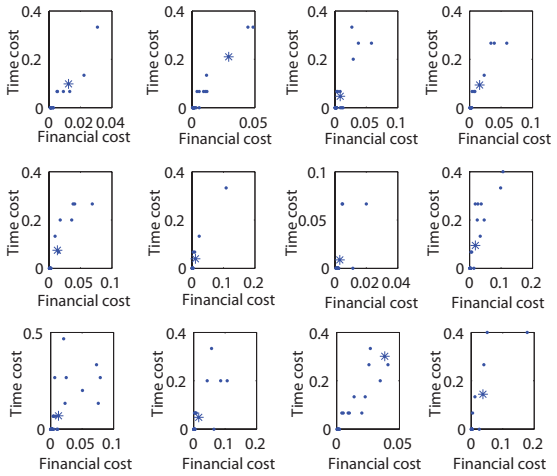

Figure 10: An Illustration of Gaussian Parameters of User Cost.

4.7 The Model Efficiency

In this subsection, we compare the efficiency of four models. Table 2 shows the training time of GcPMF, cPMF, PMF and SVD models. Here, we used the same 10-dimensional latent features. Since there is some additional cost for computing the similarity function in GcPMF and cPMF, a little more time is required for each updating in GcPMF and cPMF than the competing models, such as PMF and SVD. In addition, the computing of Gaussian distribution in GcPMF is more time-consuming than the computing of Frobenius norm in cPMF. However, the computing time of GcPMF is still linearly increasing as the number of training pairs increases as discussed in Subsection 3.3.

Table 2: A Comparison of the Model Efficiency

Models	Training Time (Second)
SVD	3.623112
PMF	3.411250
cPMF	4.894951
GcPMF	10.878244

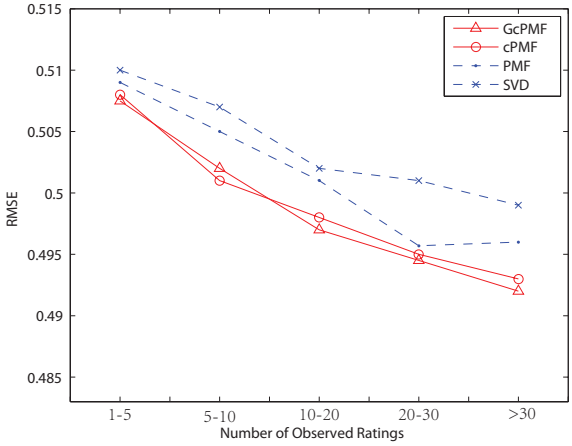

Figure 11: Performances on Different Users.

5. CONCLUSION AND DISCUSSION

In this paper, we studied the problem of travel tour recommendation by analyzing a large amount of travel logs collected from a travel agent company. One unique characteristic of tour recommendation is that there are different financial and time costs associated with each travel package. Different tourists usually have different levels of "affordability" for these two aspects of cost. Thus, we explicitly incorporated observable and unobservable cost factors into the recommendation model; that is, we used a latent factor model to represent unobserved cost factors together with other types of latent variables to model additional latent factors.

Specifically, we developed a cost-aware latent factor model, called cPMF, to learn the user/item latent features and user cost preferences simultaneously. In cPMF, we model unobserved user's cost with a 2-dimensional vector and learn the user cost vector and latent features. In addition, considering that there is usually some variance for the user cost preference, we further model user's cost with a Gaussian distribution and propose the enhanced GcPMF model. Experimental results on real-world travel logs showed that both cPMF and GcPMF models led to better learning performances than benchmark methods, such as PMF and SVD, while the learning performance of GcPMF is slightly better than that of cPMF. Furthermore, we illustrated the learned user cost preferences, which are helpful for travel companies to profile their customers.

Finally, we would like to point out that, although we have focused on incorporating costs into the travel-related applications in this paper, our approach is more general and goes well beyond travel-related applications. In particular, our approach can be applicable to a broad range of other applications in which some costs are observable and explicitly defined, while others are unobservable and need to be modeled using latent variables. For instance, let us consider an electronic product recommendation scenario, such as recommending a digital camera at Amazon.com. Different users can usually afford different prices. To provide better personalized recommendation to different users, it is expected to apply cost-aware collaborative filtering models to learn the customer's cost preference and other interests. The prediction of rating can be decided by the learned latent features and user cost preference. This will most likely lead to bet-

ter performance for recommendation than traditional models without considering the cost context. We would like to explore these issues further as a part of our future research of cost-aware recommender systems.

6. ACKNOWLEDGEMENTS

This research was partially supported by National Science Foundation (NSF) via grant number CCF-1018151, and National Natural Science Foundation of China (NSFC) via project numbers 70890082 and 71028002. Please contact Professor Hui Xiong for any comments.

7. REFERENCES

[1] R. P. Adams, G. E. Dahl, and I. Murray. Incorporating side information in probabilistic matrix factorization with gaussian processes. In *Computing Research Repository - CORR*, 2010.

[2] G. Adomavicius and A. Tuzhilin. Towards the next generation of recommender systems: A survey of the state-of-the art and possible extensions. *TKDE, 2005*, 2005.

[3] D. Agarwal and B. C. Chen. Regression-based latent factor models. In *In KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 19–28, 2009.

[4] L. Baltrunas, F. Ricci, and B. Ludwig. Context relevance assessment for recommender systems. In *Proceedings of the 2011 International Conference on Intelligent User Interfaces, 13-16 February 2011, Palo Alto, CA, USA*, 2011.

[5] R. M. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *IEEE ICDM 2007*, pages 43–52, 2007.

[6] R. D. Burke. Hybrid web recommender systems. *Lecture Notes in Computer Science*, 4321:377–408, 2007.

[7] F. Cena, L. Console, C. Gena, A. Goy, G. Levi, S. Modeo, and I. Torre. Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide. *AI Communication*, 19(4):369–384, 2006.

[8] L.-S. Chen, F.-H. Hsu, M.-C. Chen, and Y.-C. Hsu. Developing recommender systems with the consideration of product profitability for sellers. *Information Sciences*, 178(4):1032–1048, 2008.

[9] A. Das, C. Mathieu, and D. Ricketts. Maximizing profit using recommender systems. In *WWW*, 2010.

[10] M. Deshpande and G. Karypis. Item-based top-n recommendation. In *ACM Transactions on Information Systems*, pages 22(1):143–177, 2004.

[11] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, and M. J. Pazzani. An energy-efficient mobile recommender system. In *KDD 2010*.

[12] Q. Gu, J. Zhou, and C. H. Q. Ding. Collaborative filtering weighted nonnegative matrix factorization incorporating user and item graphs. In *SIAM SDM*, pages 199–210, 2010.

[13] Q. Hao, R. Cai, C. Wang, R. Xiao, J.-M. Yang, Y. Pang, and L. Zhang. Equip tourists with knowledge mined from travelogues. In *the 19th International World Wide Web Conference*, 2010.

[14] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems - TOIS*, 22(1):89–115, 2004.

[15] K. Hosanagar, R. Krishnan, and L. Ma. Recommended for you: The impact of profit incentives on the relevance of online recommendations. In *Proceedings of the International Conference on Information Systems*, 2008.

[16] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD 2008*, pages 426–434, 2008.

[17] Y. Koren. Collaborative filtering with temporal dynamics. In *KDD 2009*, pages 447–456, 2009.

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

[19] T.-K. H. J. S. J. G. C. Liang Xiong, Xi Chen. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SIAM International Conference on Data Mining*, pages 211–222, 2010.

[20] 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.

[21] Z. Lu, D. Agarwal, and I. S. Dhillon. A spatio-temporal approach to collaborative filtering. In *Conference on Recommender Systems - RecSys*, pages 13–20, 2009.

[22] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *Research and Development in Information Retrieval*, pages 203–210, 2009.

[23] B. Marlin. Modeling user rating profiles for collaborative filtering. In *In NIPS*. MIT Press, 2003.

[24] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Neural Information Processing Systems 21 (NIPS 2008)*, 2008.

[25] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl. Application of dimensionality reduction in recommender system - a case study. In *In ACM WebKDD Workshop*, 2000.

[26] N. Srebro, J. Rennie, and T. Jaakkola. Maximum margin matrix factorizations. In *Advances in Neural Information Processing Systems (NIPS) 17, 2005*.

[27] M. Wang, X.-S. Hua, J. Tang, and R. Hong. Beyond distance measurement: Constructing neighborhood similarity for video annotation. In *IEEE Transactions on Multimedia*, volume 11, 2009.

[28] M. Wang, X.-S. Hua, J. Tang, and R. Hong. Unified video annotation via multi-graph learning. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 19, 2009.

[29] G. Xue, C. Lin, Q. Yang, W. Xi, H. Zeng, Y. Yu, and Z. Chen. Scalable collaborative filtering using cluster-based smoothing. In *In Proc. of SIGIR*, pages 114–121, 2005.

[30] Z. Yu, X. Zhou, D. Zhang, C.-Y. Chin, X. Wang, and J. Men. Supporting context-aware media recommendations for smart phones. *IEEE Pervasive Computing*, 5(3):68–75, July 2006.