Travel Recommendation via Fusing Multi-Auxiliary Information into Matrix Factorization

LEI CHEN, Nanjing University of Science and Technology
ZHIANG WU* and JIE CAO, Nanjing University of Finance and Economics
GUIXIANG ZHU, Nanjing University of Science and Technology
YONG GE, University of Arizona

As an e-commerce feature, the personalized recommendation is invariably highly-valued by both consumers and merchants. The e-tourism has become one of the hottest industries with the adoption of recommendation systems. Several lines of evidence have confirmed the travel-product recommendation is quite different from traditional recommendations. Travel products are usually browsed and purchased relatively infrequently compared with other traditional products (e.g., books, food, etc.), which gives rise to the extreme sparsity of travel data. Meanwhile, the choice of a suitable travel product is affected by an army of factors such as departure, destination, financial and time budget. To address these challenging problems, in this paper, we propose a Probabilistic Matrix Factorization with Multi-Auxiliary Information (PMF-MAI) model in the context of the travel-product recommendation. In particular, PMF-MAI is able to fuse the probabilistic matrix factorization on the user-item interaction matrix with the linear regression on a suite of features constructed by the multiple auxiliary information. In order to fit for the sparse data, PMF-MAI is built by a whole-data based learning approach which utilizes unobserved data to increase the coupling between probabilistic matrix factorization and linear regression. Extensive experiments are conducted on a real-world dataset provided by a large tourism e-commerce company. PMF-MAI shows an overwhelming superiority over all competitive baselines on the recommendation performance. Also, the importance of features is examined to reveal the crucial auxiliary information having a great impact on the adoption of travel products.

CCS Concepts: • **Information systems** → **Recommender systems**; *Information retrieval*; *Retrieval tasks and goals*;

Additional Key Words and Phrases: Travel product recommendation, probabilistic matrix factorization, linear regression, multiple auxiliary information, recommender systems

ACM Reference format:
Lei Chen, Zhiang Wu, Jie Cao, Guixiang Zhu, and Yong Ge. 2019. Travel Recommendation via Fusing Multi-Auxiliary Information into Matrix Factorization. *ACM Trans. Intell. Syst. Technol.* 1, 1, Article 1 (October 2019), 24 pages.

*The corresponding author

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1000901, in part by the National Natural Science Foundation of China (NSFC) under Grant 71571093, Grant 91646204 and Grant 71801123, in part by the National Center for International Joint Research on E-Business Information Processing under Grant 2013B01035.
Author's addresses: L. Chen and G. Zhu, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China; Z. Wu, Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing, China; J. Cao, School of Information Engineering, Nanjing University of Finance and Economics, Nanjing, China; Y. Ge, Eller School of Management, University of Arizona, USA.

DOI: 0000001.0000001

DOI: 0000001.0000001

1 INTRODUCTION

The tourism industry has experienced steady growth almost every year worldwide. Sensing these huge business opportunities, more and more online travel agencies (OTA) keep popping up all across the world, *e.g.,* TripAdvisor, Expedia, Trip.com, Tuniu, etc. These online travel agencies are able to provide services including transportation ticketing, packaged tours, accommodation reservation, corporate travel management, which are usually packaged as various travel products [30]. In this context, the usage of online information has become a major trend among travelers [21], along with online reservations for travel products becoming an important application. The iResearch data* shows that the online travel booking rate has now reached over 40% with an OTA market growing to $7.69 billion in China. In response, many OTA platforms have adopted recommender systems as a marketing communication tool so as to facilitate travelers learning and purchasing their products [17]. Both OTA platforms and travelers benefit from such recommender systems. Prospective travelers can quickly locate the travel products satisfying their personalized requirements. Recommender systems also help OTA improve services, attract and retain customers, and eventually increase conversions from browsers to buyers.

In the literature, plenty of studies on tourism-oriented recommendation have been devoted to identify points of interest (POI) by regarding user's attributes and construct the personalized itinerary for recommendation [15, 19, 25], through mining various types of data, *e.g.,* GPS trajectories [47], check-in records [46], travelogues [18], geo-tagged photos [25], and so on. Most of the research has considered "Where, When, Who" issues to model user mobility. In fact, this domain closely relates to the fields of location-based social network services and urban computing [52]. Nevertheless, our study is highly-related to another research stream of literature which explores the intelligent recommendation for *travel products* (sometimes termed as *travel packages*) [9, 30, 31, 41, 53]. These studies have repeatedly verified that the recommendation of travel products is remarkably different from that of traditional items, *e.g.,* movies, books, or groceries. Specifically, the user-item interaction matrix in the context of tourism is extremely sparse, since the relatively expensive travel-products lead to infrequent browsing and purchasing. This implies that the tourism recommendation needs to exploit other rich information for enhancing its performance. Furthermore, compared with the recommendation of traditional items, the recommendation of travel products is greatly affected by contextual factors such as the departure city, the landscapes of destination, travel seasons, financial and time budget, etc.

The auxiliary information used in recommender systems generally involves user-specific features, item-specific features and global features [1, 2, 33, 35, 44, 49]. User/item-specific features provide additional description of users or items, *e.g.,* item information, user demographics. The global features are also known as dyadic features which denote a similarity degree between a user and a product [49]. To work around the conundrum of travel-product recommendation, studies have attempted to incorporate partial auxiliary information to alleviate the data sparseness problem and thus to improve the recommendation accuracy. For example, Liu *et al.* introduced a tourist-area-season topic model to integrate the descriptive text of travel products with the collaborative filtering model [30, 31]. Ge *et al.* [9] developed the cost-aware latent factor model to take both financial and time costs into account. He *et al.* [11] extended the topic model to incorporate the social influence besides the descriptive text of travel products. Liu *et al.* [28] proposed a multiple factors model for estimating passengers' future air travel pattern. The auxiliary information used

*http://www.iresearchchina.com/

in these studies are almost item-specific features. However, the global features act a fat part in the tourism-oriented recommendation, because many features make sense when they link up users with products. For instance, the city that a consumer lives in has little effect on modeling preferences, but it becomes meaningful when it relates to the departure city of a travel product. Hence, the pressing concern for travel product recommendation is the ability for fusing all-round knowledge. That is, the recommendation model should not be restricted to some specific type of auxiliary information, whereas it calls for a flexible and universal framework that is able to effectively incorporate all kinds of auxiliary information that is available from data.

In this paper, we propose a Probabilistic Matrix Factorization with Multi-Auxiliary Information (PMF-MAI) model for the travel-product recommendation. PMF-MAI is capable of fusing multiple auxiliary information into matrix factorization and utilizing the plenty of unobserved value to improve the recommendation accuracy. This paper makes three key contributions to the literature of recommender systems as well as travel recommendations:

(1) PMF-MAI is able to jointly model the multiple auxiliary information affecting the adoption of travel products and user-product interaction matrix explicitly expressing users' preferences. Although PMF-MAI is used for travel product recommendation throughout this paper, the model itself is truly universal to other recommendation or prediction problems fed with both interaction matrix and auxiliary information.
(2) PMF-MAI adopts a whole-data based learning framework working on both observed and unobserved samples. By sufficiently exploiting the unobserved data, PMF-MAI is conducive to alleviate the sparsity of user-product interaction data.
(3) PMF-MAI is evaluated on a real-life dataset obtained from a large tourism e-commerce company in China. A set of dyadic features are constructed in the context of e-tourism. Experimental results not only demonstrate that PMF-MAI outperforms several competitive baselines for the travel product recommendation, but also validate the effectiveness of the proposed dyadic features.

Organization: The remainder of this paper is organized as follows. In Section 2, we summarize the related work in detail. In Section 3, we begin by describing the problem that we study in this article, and then present our proposed recommendation framework PMF-MAI. Section 4 introduces the real-life dataset and the construction of features by using the multi-auxiliary information. We exhibit the experimental results in Section 5 and finally conclude this paper in Section 6.

2 RELATED WORK

In this section, we survey the relevant literature in two streams of research: tourism-oriented recommendations, and matrix factorization (MF) based recommendation methods.

2.1 Tourism-Oriented Recommendations

Here, we discuss two substreams related to the study of tourism-oriented recommendations. The first one relates to predicting next location (*i.e.,* POI) by regarding interest preferences, and further to generating itinerary as a sequence of locations under trip constraints (*e.g.,* time limits, start and end points). Since a sequence of POI visits is naturally interpreted as a session [36], the session-based recommendation methods are adopted for predicting the next POI [7, 48]. These methods learn user transactional behavioral patterns (*e.g.,* sequential patterns) and the user preference shift from one transaction to another for recommendations. In addition to the user-location data, plenty of contextual information has been exploited to improve the recommendation accuracy. Cheng *et al.* [3] proposed a Gaussian mixture model for taking social influence and geographical information into consideration. Similarly, Liu *et al.* [27] modeled the geographical influences and some other factors

by a general geographical probabilistic factor model. On the other hand, tour recommendation and itinerary planning depend on the combination of various factors such as POI popularity and category, trip constraints, and interest preferences, which are usually approached as an optimization problem [19, 25, 26, 46]. For instance, the variants of traveling salesman and orienteering problem are two widely-used optimization models in the field of tour recommendations.

Although a wide array of studies fall within the aforementioned field, our work is highly-related to the second substream: travel product or package recommendations. Much of the available literature on this research substream used a dataset provided by an offline travel company, consisting of tens of thousands of expense records between users and travel packages [8, 9, 11, 30, 31, 41]. Compared with traditional products, a notable feature of the travel-package recommendation is that the user-product interaction matrix is overly sparse and there exists amounts of auxiliary information that is potentially useful to an effective recommendation. Along this line, information about area and season is extracted from the descriptive text of each travel package, and a hybrid recommendation method that has the ability to combine many constraints is developed [30, 31]. Likewise, Ge *et al.* [8, 9] examined the effect of both financial and time cost on travel product purchases and presented two kinds of cost-aware recommendation models. He *et al.* [11] considered the social influence of co-travelers to enhance the representation of travel interests. Recently, Liu *et al.* [28] presented a topic model fusing multiple factors such as gender, age, the customer similarity graph to predict customer airline travel preferences. However, previous studies on this regard focused on exploiting some specific type of factor to improve the recommendation quality, little work has considered designing a systematic and flexible framework to incorporate all-round knowledge for travel-product recommendations, leaving an open field worthy of research. Furthermore, our work is one of few studies to investigate the travel-product recommendation problem on the real-life data sourced from an OTA platform, which is particularly important for practitioners as the OTA platform has become common among online retailers.

2.2 MF-based Recommendation Methods

Matrix factorization (MF) aims to find two or more matrices such that their product can well approximate the original data matrix [34], and it has been successfully applied to handle a bank of recommendation problems. Historically, many researchers have studied how to effectively leverage auxiliary information (*e.g.,* social relations among users [3, 6, 42] and geographical information [24, 27, 51]) and incorporate them into MF-based recommendation models. From a technical perspective, two principled methods are noteworthy. The first one is the so-called matrix co-factorization [3, 27, 42, 51] that simultaneously decomposes two or three matrices with the share of the latent factor matrices. Although this approach is very insightful, the extension to incorporate multiple kinds of auxiliary information usually represented as multiple matrices is largely neglected. The other one is the regression-based latent factor model, where multiple auxiliary information can be encoded as features and fed together with user-item interaction matrix by using the linear or non-linear regression models [1, 6, 24, 33, 39]. Compared with co-factorization, the regression-based latent factor method is more reasonable and flexible to fuse multiple auxiliary information. Agarwal and Chen [1] assumed that the latent factor matrices are generated from the side information via linear regression and their product should be approximated with the original data matrix. Park *et al.* [33] proposed the Bayesian matrix factorization approach to alleviate the overfitting the problem of the traditional regression-based latent factor models. Chen *et al.* [2] simultaneously incorporated features and past user-item interactions through a generalized linear model and developed a machine learning toolkit for feature-based matrix factorization. However, these above methods either ignore global features or only regard global features as the bias term. Global features have a

great influence on the tourism-oriented recommendation, which directly affects users' preferences for products. Different from the existing models, we fuse the global features via linear regression and minimize the deviation between matrix factorization and linear regression. Meanwhile, we treat the regressions of global features as the calibration of factorization on *unobserved values*. In fact, our PMF-MAI model provides a reasonable and effective way to extend the regression-based MF model to learn on both observed and unobserved data.

Recent research has further improved MF-based recommendation models by focusing on the usage of the missing data, *i.e.,* unobserved values [4, 13, 24, 43]. For instance, Volkovs *et al.* [43] presented the SVD block-factorization approach that enables SVD to handle the missing data. Devooght *et al.* [4] offered an interesting approach where the unobserved ratings are modeled as a prior estimate that is dealt with separately from the observed ratings. They showed that to make MF-based models learning on unobserved values is very critical to enhance the recommendation performance on the sparse data. Nevertheless, much of the research up to now regards the unobserved values as the negative feedback, which is not always consistent with the reality.

Despite previous works have made significant improvements on recommendation performances, the newly-proposed PMF-MAI model has its distinctive characteristics and advantages. Firstly, PMF-MAI is a systematic and scalable approach that is capable of fusing multiple sources of auxiliary information, which is superior to most current models that fail to fully fuse global features. Secondly, PMF-MAI fully exploits unobserved values as the calibration of probabilistic matrix factorization with linear regression, which is more suitable for handing highly sparse data.

3 THE PMF-MAI MODEL

In this section, we first outline the travel-product recommendation problem to be studied. Then, we present the recommendation model named Probabilistic Matrix Factorization with Multi-Auxiliary Information (PMF-MAI), which provides an integrated framework for fusing the probabilistic matrix factorization on user-item interaction matrix and the linear regression on a set of features constructed by multi-auxiliary information.

Throughout the paper, lowercase symbols (such as a, b) denote scalars, bold lowercase symbols (such as $\mathbf{a}$, $\mathbf{b}$) represent vectors, bold uppercase symbols (such as $\mathbf{A}$, $\mathbf{B}$) denote matrices and calligraphy symbols (such as $\mathcal{A}$, $\mathcal{B}$) represent tensors. For better illustration, Table 1 lists all mathematical notations used in this paper.

3.1 Problem Statement

In the literature, the recommendation task is usually specified as: given an $N \times M$ matrix $\mathbf{X}$ describing the preferences of N users over M items, we aim to recommend each user with a set of new items that this user might be interested in but has never been keen on these items before. The matrix $\mathbf{X}$ has various definitions in different scenarios. For example, $\mathbf{X}$ can represent the browsing behavior or consumption behavior on e-commerce sites [22], and can also denote the five-grade ratings in the classic movie recommendation. The recommendation task is then equivalent to the prediction for missing values of $\mathbf{X}$, and thus the recommended items are generated by the ranking of predicted values.

In many real applications, there is multiple auxiliary information producing effects on users' interests. Taking the application to be addressed throughout this paper as an example: the recommendation of travel products on an OTA platform. We can collect the frequency that a user has clicked, *i.e.,* browsed, a web page about one travel product, which is naturally placed into the matrix $\mathbf{X}$. This matrix only indicates users' interests against the destinations or landscapes of different travel products. However, besides the users' historical interests, multiple other auxiliary

Table 1. Mathematical Notations

Notation	Description
N	number of instances, *e.g.,* users, sessions
M	number of items, *i.e.,* travel products
K	dimension of latent factors
D	number of features constructed by auxiliary information
$\mathbf{X}$	user-product browse frequency matrix
$\mathbf{H}$ ($\bar{\mathbf{H}}$)	indicator matrix of observed (unobserved) values in $\mathbf{X}$
$\mathbf{U}$ ($\mathbf{u}_i$)	user latent factors (for ith user)
$\mathbf{V}$ ($\mathbf{v}_j$)	product latent factors (for jth product)
$\mathcal{Y}$	feature tensor
$\mathbf{y}_{ij}$ (y_{ijd})	(i,j) fiber of tensor $\mathcal{Y}$ (for dth feature)
$\boldsymbol{\beta}$ (β_d)	regression coefficients vector (for dth feature)

information is very important to determine whether a user prefers one travel product. For instance, most users expect to travel from home, *i.e.,* the starting place should be as near as the user's city. Furthermore, most tourists will not frequently browse some travel products of which the financial costs they are unable to afford. In other words, both time and price of a travel product should be in line with users' estimates.

Based on the above analysis, the problem discussed in this paper is how to fuse all-around auxiliary information for enhancing the prediction of missing values inside the user-item matrix $\mathbf{X}$. In other words, we target at developing a novel recommendation model that can jointly combine the interest expressed by $\mathbf{X}$ and various auxiliary information affecting the users' interests. With notations shown in Table 1, the problem discussed in this article is described as follows:

Definition 3.1 (Problem Statement). Given the partially observed preference matrix $\mathbf{H} \odot \mathbf{X}$ and the feature tensor $\mathcal{Y}$, we target at estimating the unknown preference of every instance to each product, such that we can make recommendations to the users.

3.2 Probabilistic Matrix Factorization on User-Item Matrix

Let $\mathbf{X} = [x_{ij}]_{N\times M}$ be the matrix representing the interest of every user over all items, $\mathbf{U} \in \mathbb{R}^{K\times N}$ and $\mathbf{V} \in \mathbb{R}^{K\times M}$ be two projection matrices on the latent space for users and items respectively, with column vectors $\mathbf{u}_i$ and $\mathbf{v}_j$ representing the K-dimensional user-specific and item-specific latent vectors. We decompose $\mathbf{X}$ as the product of two matrices on the joint latent space with dimension $K \ll \min(N, M)$ by using the probabilistic matrix factorization:

$$\mathbf{X} = \mathbf{U}^\top \mathbf{V} + \mathbf{E}_1, \tag{1}$$

where $\mathbf{E}_1$ is an error matrix of which each element is often modeled as a Gaussian observation noise [37], denoted as $\mathcal{N}(0, \sigma_{X1}^2)$. Since the matrix $\mathbf{X}$ is usually very sparse, an indicator matrix $\mathbf{H} \in \mathbb{R}^{N\times M}$ is adopted to indicate the observed values, of which each element $h_{ij} = 1$ indicates the observed values, otherwise $h_{ij} = 0$ for unobserved values. We then define the conditional distribution with respect to all observed values of $\mathbf{X}$ as

$$P(\mathbf{X}|\mathbf{U}, \mathbf{V}, \sigma_{X1}^2) = \prod_{i=1}^{N} \prod_{j=1}^{M} [\mathcal{N}(x_{ij}|\mathbf{u}_i^\top \mathbf{v}_j, \sigma_{X1}^2)]^{h_{ij}}. \tag{2}$$

We also place zero-mean spherical Gaussian priors [37] on user and item latent vectors respectively:

$$P(\mathbf{U}|\sigma_U^2) = \prod_{i=1}^{N} \mathcal{N}(\mathbf{u}_i|\mathbf{0}, \sigma_U^2\mathbf{I}),\ P(\mathbf{V}|\sigma_V^2) = \prod_{j=1}^{M} \mathcal{N}(\mathbf{v}_j|\mathbf{0}, \sigma_V^2\mathbf{I}). \tag{3}$$

With the Bayesian theorem, we have:

$$P(\mathbf{U}, \mathbf{V}|\mathbf{X}, \sigma_{X1}^2, \sigma_U^2, \sigma_V^2) \propto P(\mathbf{X}|\mathbf{U}, \mathbf{V}, \sigma_{X1}^2) P(\mathbf{U}|\sigma_U^2) P(\mathbf{V}|\sigma_V^2). \tag{4}$$

By putting Eqs. (2) and (3) into Eq. (4), we can obtain the log of the posterior distribution over user and item latent vectors with respect to the prior variance and observation noise variance.

$$\begin{aligned} \log P(\mathbf{U}, \mathbf{V}|\mathbf{X}, \sigma_{X1}^2, \sigma_U^2, \sigma_V^2) \propto & -\frac{1}{2\sigma_{X1}^2} \sum_{i=1}^{N} \sum_{j=1}^{M} h_{ij}(x_{ij} - \mathbf{u}_i^\top \mathbf{v}_j)^2 \\ & -\frac{1}{2\sigma_U^2} \sum_{i=1}^{N} \mathbf{u}_i^\top \mathbf{u}_i - \frac{1}{2\sigma_V^2} \sum_{j=1}^{M} \mathbf{v}_j^\top \mathbf{v}_j. \end{aligned} \tag{5}$$

According to Eq. (5), to compute the maximum a posteriori (MAP) estimation of $\mathbf{U}$ and $\mathbf{V}$ is equivalent to minimize the following objective function denoted as $\mathcal{J}_1$.

$$\mathcal{J}_1 = \frac{1}{\sigma_{X1}^2} ||\mathbf{H} \odot (\mathbf{X} - \mathbf{U}^\top \mathbf{V})||_F^2 + \frac{1}{\sigma_U^2} ||\mathbf{U}||_F^2 + \frac{1}{\sigma_V^2} ||\mathbf{V}||_F^2, \tag{6}$$

where $||\cdot||_F^2$ denotes the Frobenius norm and $\odot$ is the Hardamard product.

Remark: The SVD-based models are the most widely used in recommender systems [20, 32]. They minimize the Frobenius norm between the preference matrix and the SVD approximation. Whereas PMF only optimizes the reconstruction error, which makes it far more flexible. For this reason, we select PMF as the basis model which is easily extended to incorporate multi-auxiliary information and to support whole-data based learning on unobserved values.

3.3 Linear Regression for Multi-Auxiliary Information

The auxiliary information that affects the recommendation is data-specific. Nevertheless, we propose to model this auxiliary information as a set of pre-defined features associated with each element $x_{ij} \in \mathbf{X}$ to be predicted. For instance, features constructed by auxiliary information associated a pair of user and travel-product, *i.e.,* x_{ij}, may contain the distance between user's departure city and landscapes included in this product and the price utility of the user with respect to this product. The feature construction in the case of travel products recommendation will be introduced in Section 4.2. Without loss of generality, for any $x_{ij} \in \mathbf{X}$, we assume it is associated with a $(D-1)$-dimensional feature vector $\mathbf{y}_{ij} = [y_{ijd}]_{(D-1)\times 1}$. Then, if we regard x_{ij} as the response and $\mathbf{y}_{ij}$ as the control variables, the regression function is

$$x_{ij} = \boldsymbol{\beta}^\top \mathbf{y}_{ij} + \beta_D, \tag{7}$$

where $\boldsymbol{\beta} = [\beta_d]_{(D-1)\times 1}$ denotes the regression coefficients corresponding to every feature. For simplicity, if we set $y_{ijD} = 1$, Eq. (7) can be written as $x_{ij} = \boldsymbol{\beta}^\top \mathbf{y}_{ij} = \sum_{d=1}^{D} \beta_d y_{ijd}$.

We regard $\mathbf{y}_{ij}$ as the (i, j) fiber of tensor $\mathcal{Y}$, then the Eq. (7) can be further rewritten as:

$$\mathbf{X} = \mathcal{Y}_{\times d} \boldsymbol{\beta} + \mathbf{E}_2,$$

where $\times d$ is the d-mode product between tensor $\mathcal{Y}$ and vector $\boldsymbol{\beta}$. Again we adopt a Gaussian observation noise with variance σ_{X2}^2 to model the error matrix $\mathbf{E}_2$. Similar to Eq. (2), the conditional distribution over observed values in $\mathbf{X}$ is

$$P(\mathbf{X}|\boldsymbol{\beta}, \sigma_{X2}^2) = \prod_{i=1}^{N} \prod_{j=1}^{M} [\mathcal{N}(x_{ij}|\boldsymbol{\beta}^\top \mathbf{y}_{ij}, \sigma_{X2}^2)]^{h_{ij}}. \tag{8}$$

Likewise, we exploit zero-mean spherical Gaussian prior on the regression weight vector:

$$P(\boldsymbol{\beta}|\sigma_{B1}^2) = \prod_{d=1}^{D} \mathcal{N}(\beta_d|\mathbf{0}, \sigma_{B1}^2 \mathbf{I}).$$

According to the Bayesian theorem, we have

$$P(\boldsymbol{\beta}|\mathbf{X}, \sigma_{X2}^2, \sigma_{B1}^2) \propto P(\mathbf{X}|\boldsymbol{\beta}, \sigma_{X2}^2) P(\boldsymbol{\beta}|\sigma_{B1}^2). \tag{9}$$

The log of the posterior distribution in Eq. (9) is given by

$$\log P(\boldsymbol{\beta}|\mathbf{X}, \sigma_{X2}^2, \sigma_{B1}^2) \propto -\frac{1}{2\sigma_{X2}^2} \sum_{i=1}^{N} \sum_{j=1}^{M} h_{ij} (x_{ij} - \boldsymbol{\beta}^\top \mathbf{y}_{ij})^2 - \frac{1}{2\sigma_{B1}^2} \sum_{d=1}^{D} \beta_d^2. \tag{10}$$

Therefore, the MAP estimation of $\boldsymbol{\beta}$ is equivalent to minimize the objective function $\mathcal{J}_2$ as follows.

$$\mathcal{J}_2 = \frac{1}{\sigma_{X2}^2} ||\mathbf{H} \odot (\mathbf{X} - \mathcal{Y}_{\times d} \boldsymbol{\beta})||_F^2 + \frac{1}{\sigma_{B1}^2} ||\boldsymbol{\beta}||_F^2. \tag{11}$$

Remark: Much of prior work [9, 24] only considered one type of feature constructed by auxiliary information that was usually represented as a matrix. Thus, the matrix factorization approach can be directly utilized to obtain the user/item latent vector on the feature matrix. In this paper, we scale the feature matrices to a tensor for incorporating richer auxiliary information. Hence, we adopt the regression model to handle such a complex case, because the matrix factorization is ineffective to multi-dimensional features. If only one feature is considered, *i.e.,* $D - 1 = 1$, our model is approximately reduced to many of models in previous studies [9, 24], except that the linear regression instead of matrix factorization is utilized in our model.

3.4 Modeling Unobserved Values

Up to now, we have considered the observed value in $\mathbf{X}$, constrained by the indicator variable H_{ij}, in both probabilistic matrix factorization model and linear regression model. However, lack of consideration for unobserved data is likely to increase the bias of the maximum likelihood inference [4]. Therefore, we propose to use the auxiliary information to calibrate the PMF model on unobserved values. To be specific, we attempt to minimize the bias between the PMF model and the linear regression on tensor of features constructed by auxiliary information:

$$\mathbf{U}^\top \mathbf{V} = \mathcal{Y}_{\times d} \boldsymbol{\beta} + \mathbf{E}_3.$$

Along this line, the conditional distribution to indicate the Gaussian noise over unobserved values is

$$P(\mathbf{U}, \mathbf{V}|\boldsymbol{\beta}, \sigma_{B2}^2) = \prod_{i=1}^{N} \prod_{j=1}^{M} [\mathcal{N}(\mathbf{u}_i^\top \mathbf{v}_j|\boldsymbol{\beta}^\top \mathbf{y}_{ij}, \sigma_{B2}^2)]^{\bar{h}_{ij}}. \tag{12}$$

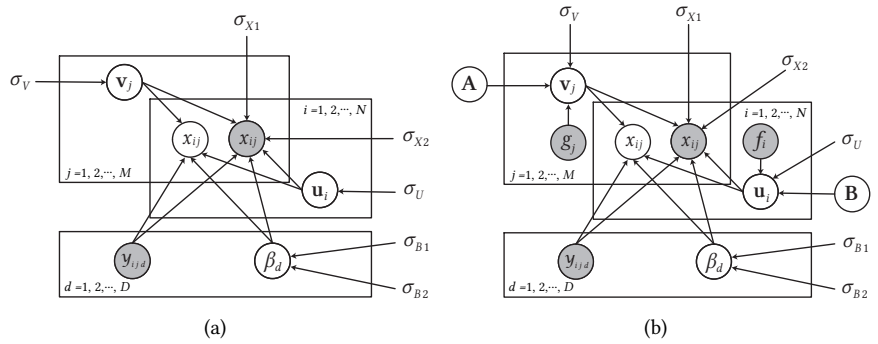

Fig. 1. Graphical representation of the PMF-MAI model. Figure (a) shows the basic version of PMF-MAI; Figure (b) shows the variant of PMF-MAI incorporating user/item-specific features.

where $\bar{h}_{ij}$ is the negation of $h_{i,j}$, *i.e.,* $\bar{h}_{ij} = 1 - h_{ij}$. The log of the posterior probability can be computed as follows:

$$\log P(\mathbf{U}, \mathbf{V} | \boldsymbol{\beta}, \sigma_{B2}^2) \propto -\frac{1}{2\sigma_{B2}^2} \sum_{i=1}^{N} \sum_{j=1}^{M} \bar{h}_{ij} (\mathbf{u}_i^\top \mathbf{v}_j - \boldsymbol{\beta}^\top \mathbf{y}_{ij})^2. \tag{13}$$

Thus, the MAP estimation is to minimize objective function $\mathcal{J}_3$ as:

$$\mathcal{J}_3 = \frac{1}{\sigma_{B2}^2} ||\bar{\mathbf{H}} \odot (\mathcal{Y}_{\times d} \boldsymbol{\beta} - \mathbf{U}^\top \mathbf{V})||_F^2. \tag{14}$$

Remark: Since the user-item interaction matrices would be very sparse in practice, the matrix factorization must be performed despite missing data (*i.e.,* unobserved values), a problem which has seen significant research attention. Most existing techniques [4, 13, 24] treated unobserved values as negative examples and utilized the unknown rating, which was usually set as the worst rating (*e.g.,* 0), to guide the matrix factorization on missing data. However, we argue that the unobserved values are not always appearing in products that users dislike. That is why we propose to optimize the error between the output of PMF, *i.e.,* $\mathbf{U}^\top \mathbf{V}$, and the linear regression on features, *i.e.,* $\mathcal{Y}_{\times d} \boldsymbol{\beta}$.

3.5 Integrated Model: PMF-MAI

In order to provide more accurate and efficient model, we integrate the above-mentioned three objective functions, *i.e.,* $\mathcal{J}_1$, $\mathcal{J}_2$ and $\mathcal{J}_3$, to get the joint model PMF-MAI. Specifically, we utilize the linear weighting method to convert the problem of multi-objective optimization into a mono-objective one. To better understand PMF-MAI, we display the graphical representation of PMF-MAI in Fig. 1(a). In the graphical representation, white and gray circles represent hidden and seen circles, respectively. The directed edges between variables indicate dependencies between the variables. Each plate represents a group of variables. In our model, the top plate, middle plate and bottom plate represent all the variables related to a specific product, user and feature. And we repeat the generation process by M, N and D times, respectively. As we can see, based on whether the data is observed or not, x_{ij} is divided into two categories. The unobserved x_{ij} is used as the calibration of regression model with matrix factorization. The objective function of PMF-MAI is defined as

$$\mathcal{J} = \alpha_1 \mathcal{J}_1 + \alpha_2 \mathcal{J}_2 + \alpha_3 \mathcal{J}_3,$$

Algorithm 1 PMF-MAI: the Unified Procedure

Require: Matrices $\mathbf{X}$, $\mathbf{H}$; Tensor $\mathcal{Y}$; Dimension of latent factors K; Hyperparameters λ_{X1}, λ_U, λ_V, λ_{B2}, λ_{X2}, λ_{B1};
Ensure: Predicted matrix $\widehat{\mathbf{X}}$;

procedure PMF-MAI($\mathbf{X}, \mathbf{H}, \mathcal{Y}, K$)
  Initialize $\mathbf{U}$, $\mathbf{V}$ and $\boldsymbol{\beta}$ with random numbers within the range $(0, 1)$;
  Perform random sampling on unobserved values of $\mathbf{X}$; ▷ optional step
  while not converged **do**
    Update $\mathbf{U} \leftarrow \mathbf{U} - \xi \frac{\partial \mathcal{J}}{\partial \mathbf{U}}$ with Eq. (16);
    Update $\mathbf{V} \leftarrow \mathbf{V} - \xi \frac{\partial \mathcal{J}}{\partial \mathbf{V}}$ with Eq. (17);
    Update $\beta_d \leftarrow \beta_d - \xi \frac{\partial \mathcal{J}}{\partial \beta_d}$ with Eq. (18);
  end while
  return $\widehat{\mathbf{X}} = \mathbf{U}^\top \mathbf{V}$;
end procedure

where α_1, α_2 and α_3 are weights for balancing three objective functions, and $\alpha_1 + \alpha_2 + \alpha_3 = 1$. By putting Eqs. (6), (11) and (14) together, we have:

$$\begin{aligned}\mathcal{J} = &\frac{\lambda_{X1}}{2}||\mathbf{H} \odot (\mathbf{X} - \mathbf{U}^\top \mathbf{V})||_F^2 + \frac{\lambda_{X2}}{2}||\mathbf{H} \odot (\mathbf{X} - \mathcal{Y}_{\times d}\boldsymbol{\beta})||_F^2 \\ &+ \frac{\lambda_{B2}}{2}||\bar{\mathbf{H}} \odot (\mathcal{Y}_{\times d}\boldsymbol{\beta} - \mathbf{U}^\top \mathbf{V})||_F^2 + \frac{\lambda_U}{2}||\mathbf{U}||_F^2 \\ &+ \frac{\lambda_V}{2}||\mathbf{V}||_F^2 + \frac{\lambda_{B1}^2}{2}||\boldsymbol{\beta}||_F^2, \end{aligned} \tag{15}$$

where $\lambda_{X1} = \frac{2\alpha_1}{\sigma_{X1}^2}, \lambda_{X2} = \frac{2\alpha_2}{\sigma_{X2}^2}, \lambda_{B2} = \frac{2\alpha_3}{\sigma_{B2}^2}, \lambda_U = \frac{2\alpha_1}{\sigma_U^2}, \lambda_V = \frac{2\alpha_1}{\sigma_V^2}, \lambda_{B1} = \frac{2\alpha_2}{\sigma_{B1}^2}$.

A local minimum of the objective function given by Eq. (15) can be obtained by performing gradient descent in $\mathbf{U}$, $\mathbf{V}$ and β_d, respectively.

$$\frac{\partial \mathcal{J}}{\partial \mathbf{U}} = \lambda_{X1}[\mathbf{V}(\mathbf{H}^\top \odot (\mathbf{V}^\top \mathbf{U} - \mathbf{X}^\top))] + \lambda_{B2}[\mathbf{V}(\bar{\mathbf{H}}^\top \odot ((\mathcal{Y}_{\times d}\boldsymbol{\beta})^\top - \mathbf{V}^\top \mathbf{U}))] + \lambda_U \mathbf{U}, \tag{16}$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{V}} = \lambda_{X1}[\mathbf{U}(\mathbf{H} \odot (\mathbf{U}^\top \mathbf{V} - \mathbf{X}))] + \lambda_{B2}[\mathbf{U}(\bar{\mathbf{H}} \odot (\mathcal{Y}_{\times d}\boldsymbol{\beta} - \mathbf{U}^\top \mathbf{V}))] + \lambda_V \mathbf{V}, \tag{17}$$

$$\frac{\partial \mathcal{J}}{\partial \beta_d} = \lambda_{X2} \sum_{i=1}^{N} \sum_{j=1}^{M} [h_{ij}(\mathbf{y}_{ij}^\top \boldsymbol{\beta} - x_{ij}) y_{ijd}] + \lambda_{B2} \sum_{i=1}^{N} \sum_{j=1}^{M} [\bar{H}_{ij}(\mathbf{u}_i^\top \mathbf{v}_j - \mathbf{y}_{ij}^\top \boldsymbol{\beta}) y_{ijd}] + \lambda_{B1} \beta_d. \tag{18}$$

With Eqs (16), (17) and (18), we can employ the gradient descent method to solve our PMF-MAI model. For each iteration, we update $\mathbf{U} = \mathbf{U} - \xi \frac{\partial \mathcal{J}}{\partial \mathbf{U}}$, $\mathbf{V} = \mathbf{V} - \xi \frac{\partial \mathcal{J}}{\partial \mathbf{V}}$ and $\beta_d = \beta_d - \xi \frac{\partial \mathcal{J}}{\partial \beta_d}$, where the step size ξ is set to 0.01 in our experiments. Finally, we obtain the predicted values by $\widehat{\mathbf{X}} = \mathbf{U}^\top \mathbf{V}$. To better understand PMF-MAI, we summarize the computational procedure of PMF-MAI in Algorithm 1.

Complexity: Since our PMF-MAI has taken all unobserved values into account, the cost of updating $\mathbf{U}$ and $\mathbf{V}$ by Eqs. (16) and (17) is $O((K + D)NM)$, and the cost of updating $\boldsymbol{\beta}$ by Eq. (18) is $O(KDNM)$. Hence, the total computational complexity of PMF-MAI is $O(KDNM)$ for each iteration. Intuitively, the number of observed values $|\mathbf{X}|$ is very small in comparison to the number

of unobserved values ($NM - |\mathbf{X}|$). To reduce the computational cost of PMF-MAI, we can perform the random sampling, a commonly-used technique in handling missing data [4, 43], on unobserved before training gradient against variables (see line 3 of Algorithm 1). As a result, if we denote $\gamma \in [0, 1]$ as the sampling ratio, the total cost of PMF-MAI can be reduced to $O(KD(|\mathbf{X}| + \gamma(NM - |\mathbf{X}|)))$. We will show the effect of γ in the experimental section and demonstrate when $\gamma = 0.3$ PMF-MAI can achieve the satisfactory performance.

3.6 Connections to Existing Models

Here, we show the connections between PMF-MAI and previous models taking user/item-specific features into account, and the distinctions between PMF-MAI and previous models taking global features into account. When given auxiliary information in the form of feature vectors related with user and item, there exist a number of matrix factorization models making user of user/item feature vectors to derive latent vectors. Mathematically, let $\mathbf{F} = [f_1, \cdots, f_N]$ and $\mathbf{G} = [g_1, \cdots, g_M]$ denote feature matrices, where f_i and g_j are feature vector related with user i and item j respectively. The first class of methods is to use the *matrix co-factorization* to compute the latent vectors [40]:

$$\mathbf{F} = \mathbf{A}^\top \mathbf{U} + \mathbf{E}_F, \mathbf{X} = \mathbf{U}^\top \mathbf{V} + \mathbf{E}_X, \mathbf{G} = \mathbf{B}^\top \mathbf{V} + \mathbf{E}_G, \tag{19}$$

where $\mathbf{E}_F$, $\mathbf{E}_X$, $\mathbf{E}_G$ are the Gaussian noise. The second class is the *regression-based latent factor model* (RLFM) [1]

$$\mathbf{U} = \mathbf{A}^\top \mathbf{F} + \mathbf{E}_U, \mathbf{V} = \mathbf{B}^\top \mathbf{G} + \mathbf{E}_V, \mathbf{X} = \mathbf{U}^\top \mathbf{V}. \tag{20}$$

Many studies have adopted other generative models as the alternative of the linear regression used by RLFM such as Bayesian matrix factorization with side information (BMFSI) [35] and hierarchical BMFSI [33]. Thus far, PMF-MAI does not consider user/item-specific features and its basic version employs the probabilistic matrix factorization to obtain user/item latent vectors as shown in Eq. (1). Nevertheless, our PMF-MAI can readily be extended to incorporate user/item-specific features by replacing the basic probabilistic matrix factorization with aforementioned latent factor models. This upgrade is unlikely to alter the way that we handle global features. Fig. 1(b) shows the graphical representation of the extended PMF-MAI model incorporating user/item-specific features.

Several studies [1, 2] have considered a fraction of global features such as day-of-week about rating time, last purchase frequency associated with user and product, etc. Since bits of global features are unlikely to have a great impact to predicted values, existing methods such as RLFM [1] and SVDFeature [2] have modeled the global features as the bias term of the score function. In contrast to these methods, our PMF-MAI approach minimizes the loss between latent factor model and linear regression of global features over both observed and unobserved values and it in fact amplifies the effect of global features. This treatment is potentially effective to cope with the challenging problem raised from the tourism domain: the user-product preference matrix is very sparse and the contextual information is extremely vital to the preference modeling.

4 DATA AND FEATURES

In this section, we first introduce a real-life dataset provided by an e-travel company in China and then describe several features constructed by auxiliary information. The set of features is used to constitute the feature vector $\mathbf{y}_{ij}$ as introduced in Section 3.3.

4.1 Data Description

Our dataset is provided by Tuniu*, a large tourism e-commerce company in China. This dataset is mainly made up of web server logs for recording a series of page views of users, where only pages

*http://www.tuniu.com/

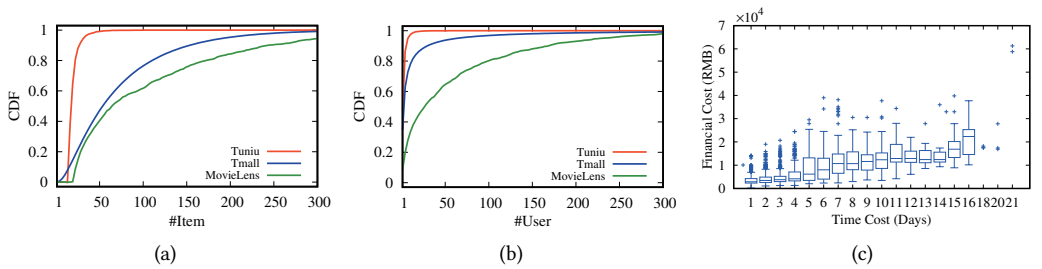

Fig. 2. Characteristics of Tuniu data. Figures (a) and (b) show the comparison of data sparseness among three real-world datasets. Figure (c) shows the distribution of financial costs of products with the same time span.

associated with travel products are maintained. Each page contains the following auxiliary information: *Page_ID*, *Departure*, *Destination*, *Price*, *Time_Span*, where *Departure* (*Destination*) represents the starting (destination) city of a travel product and *Price* (*Time_Span*) denotes the financial cost (time cost) of this travel product.

A sequence of successive records of a user is divided into a number of sessions of which each includes the sequential web pages clicked by a user during a certain period. To enrich the auxiliary information of every session, we collect sessions that arrive the website through advertising campaigns, internal search and external search engines. For these sessions, we can obtain a *Keyword* attribute that usually contains the intended destination, which will facilitate the construction of several features. In addition, the IP address of each session is also analyzed to get the location of this session denoted as *IP_City*.

We finally extracted 2,033 sessions over 15,491 pages from the logs from July to August, 2013. Then, a 2,033×15,491 matrix $\mathbf{X}$ is constructed, where each element x_{ij} denotes the count of ith session has visited jth page. The click count x_{ij} is in the range $[1, 26]$ with the average value 1.25. Moreover, the matrix $\mathbf{X}$ has only 50,533 non-zero elements, with a very low density 0.128%, which is much lower than those of traditional datasets used in recommendation. Figures 2(a) and (b) compare the data sparseness between Tuniu dataset and two typical datasets used in the recommendation area. One is the standard `MovieLens-10K`, and the other one is `Tmall` [54]. The cumulative distribution function (CDF) is the probability that takes a value less or equal to the corresponding x-value. We can see from Figure 2(a) that over 90% users in Tuniu data has clicked less than 50 items, whereas roughly 40% users of both MovieLens and Tmall have clicked and rated less than 50 items. A similar observation can be seen from Figure 2(b): the number of users that clicked each item is much more scarce in the Tuniu dataset. In addition, Figure 2(c) shows the boxplots of a group of travel products with the same time span but different financial cost. As can be seen, the product with longer time span tends to be sold in higher price, but the price fluctuation of products with the same time span is remarkably wild. This implies that both time cost and financial cost should be considered simultaneously for the user preferences modeling.

4.2 Feature Construction

Here, we delineate the construction of features as a part of the input for our PMF-MAI model. As described above, x_{ij} represents the relationship between a session and a page. So each feature y_{ijd} associated with x_{ij} should also be constructed between the session and the page. In what follows, we will construct two classes of features, including six features in total.

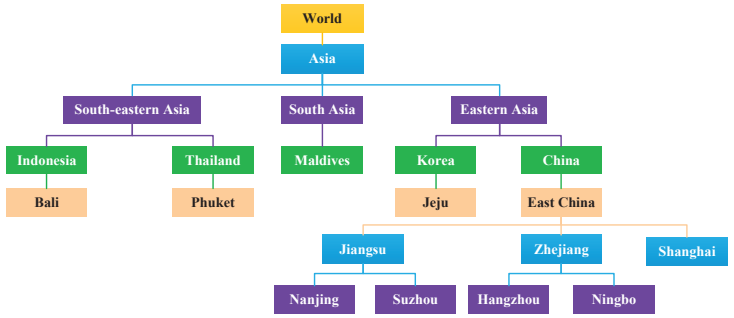

Fig. 3. A part of hierarchical structure for place names extracted from UNG.

The first class of intuitive features attempts to characterize the distance of departure city and destination city. Specifically, the locations of each session indicated by *IP_City* and the *Departure* attribute of each page are used for computing the distance of departure cities. Meanwhile, the intended destination of each session extracted from *Keyword* and the *Destination* attribute of each page are responsible for computing the distance of destination cities. We employ two kinds of methods to measure the distance between two place names: *Geographical Similarity* and *Semantic Similarity* as follows:

(1) *Geographical Similarity.* Given a pair of place names, we use Google Maps API* to compute the distance on the earth surface, denoted as $Dist(p_i, p_j)$ where p_i and p_j are two place names. Then, we utilize the Min-Max normalization to transform the distance to a geographical similarity:

$$S_1(p_i, p_j) = 1 - \frac{Dist(p_i, p_j) - \min}{\max - \min}, \tag{21}$$

where $S_1(p_i, p_j)$ represents the geographical similarity and max (min) is the maximum (minimum) value of $Dist(p_i, p_j)$.

(2) *Semantic Similarity.* In fact, the place names can be organized as a hierarchical structure, *i.e.,* a tree. For instance, one possible path of such tree is: "China→East China→Jiangsu Province→Nanjing". Thus, we use the structural similarity upon the tree to define the semantic relationship between two place names. In detail, we utilize the hierarchical structure from United Nations Geoscheme (UNG)†. Fig. 3 shows an illustrative example of this hierarchical structure. Then, we define the semantic similarity $S_2(p_i, p_j)$ as:

$$S_2(p_i, p_j) = \frac{2Depth(p_i \cap p_j)}{Depth(p_i) + Depth(p_j)}, \tag{22}$$

where $p_i \cap p_j$ denotes the last common ancestor of p_i and p_j, and $Depth(\cdot)$ gives the depth of a node in the tree, *i.e.,* the number of nodes from the root to this node. For example, if a tourist's intension destination is Thailand, according to Fig. 3, $S_2(Thailand, Phuket) = 0.889$ and $S_2(Thailand, Jeju) = 0.444$. This means this tourist has a higher preference in destination Phuket than Jeju.

By using the above two similarity measures, we can construct four features to indicate the distance of both departure and destination cities.

*https://developers.google.com/maps/
†https://en.wikipedia.org/wiki/United_Nations_geoscheme

The price and time of different travel products vary wildly. For example, travel products in our dataset are sold for dozens of dollars to thousands of dollars, and take one day to more than ten days. Hence, another class of features is designed for characterizing the preference on the financial and time cost of tourists. We adopt the method proposed by [9] to model the users' preference to financial and time cost as the Gaussian prior. Specifically, we first utilize the Min-Max method to normalize the prices of all products, *i.e.,* the *Price* attribute, and compute the average value and the standard deviation of all pages contained in a session. Then, we can obtain the probability function on price for every session by assuming the price follows 1-dimension Gaussian distribution. For each x_{ij}, the utility on price is easily given by the probability function with the price of jth page. The same process is taken on the time cost by using the *Time_Span* of every page. As a result, we construct two features to indicate both price and time preference.

5 EXPERIMENTAL RESULTS

In the following, we present our experimental setup and results. Specifically, we demonstrate: 1) the performance comparisons between PMF-MAI and other benchmark methods; 2) the understanding of features in the context of e-tourism; and 3) an analysis of parameters inside our PMF-MAI model.

5.1 Experimental Setup

All experiments were conducted on the real-world dataset as described in Section 4.1. For the 2,033×15,491 sparse matrix $\mathbf{X}$, we divided each row into a training set and a test set, by randomly extracting a certain percentage of the elements to be part of the training set and the remaining ones to be part of the test set.

5.1.1 Performance Metrics. Let $\hat{x}_{ij}$ and x_{ij} denote an estimated value and a true value respectively for a test instance. Two commonly used metrics indicating the estimated error are Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). Here, we define the average MAE and RMSE as

$$\text{MAE} = \frac{\sum_{i,j} \left| x_{ij} - \hat{x}_{ij} \right|}{n_t}, \ \text{RMSE} = \sqrt{\frac{\sum_{i,j} \left(x_{ij} - \hat{x}_{ij} \right)^2}{n_t}}, \tag{23}$$

where n_t is the number of test instances. Obviously, the small value of MAE (RMSE) indicates the better performance.

Besides metrics for estimated errors, we further employ Recall, F-measure and Normalized Discounted Cumulative Gain (NDCG) to characterize the ranking accuracy of recommendation results. Since these measures have been widely used in the literature of recommender systems [9, 30, 38, 46], we provide a very brief introduction of their calculations. In detail, the travel products in the test data are regarded as the *truly relevant items*, denoted as T_i for ith session (*i.e.,* the ith row of $\mathbf{X}$). Then, the recommendation list generated by various recommendation methods is denoted as R_i. Recall measures the ratio of the number of hits to the size of each session's test data:

$$\text{Recall} = \frac{1}{N} \sum_i \frac{|T_i \cap R_i|}{|T_i|}. \tag{24}$$

F-measure, an overall accuracy metric, is defined by the harmonic mean of precision.

$$\text{F-measure} = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}, \ \text{where Precision} = \frac{1}{N} \sum_i \frac{|T_i \cap R_i|}{|R_i|}. \tag{25}$$

NDCG [38] is the normalized position-discounted precision score:

$$\text{NDCG} = \frac{1}{N}\sum_{i}\sum_{j=1}^{|R_i|}\frac{2^{I(R_{ij}\in T_i)}-1}{\log_2^{1+j}}, \tag{26}$$

where the indicator function $I(\cdot) = 1$ if $R_{ij} \in T_i$, otherwise for 0. We further adopt a diversity metric

$$\text{Coverage} = \frac{|\bigcup_{i=1}^{N} R_i|}{M}, \tag{27}$$

and a higher coverage value indicates that the recommendation method can encompass a wider range of interests. In total, six evaluation measures are used thereafter.

5.1.2 Algorithms Compared. We consider neighborhood-based approaches (UCF and ICF), matrix factorization methods without side information (PMF and SVD), a matrix factorization method using unobserved value (eALS), a sequential pattern-based recommendation method (MCA) and matrix factorization methods with side information (RLFM, SVDFeature and BMFSI), The comparison methods are given below:

- *UCF* [16]. The standard User-based Collaborative Filtering (UCF) using Pearson correlation coefficient (PCC) as the similarity measure is used here. The number of nearest neighbors is set to 50.
- *ICF* [12]. Similar to UCF, the Item-based Collaborative Filtering (ICF) also utilizes PCC as the similarity measure, while the number of nearest neighbors is set to 200 because the number of product pages is far larger than that of sessions.
- *PMF* [29, 37, 45]. It is a standard latent factor model that is widely used in recommender systems. This can be regarded as the basic version of our PMF-MAI that does not fuse any features, that is, the estimation is performed purely based on the user-item matrix.
- *SVD* [50]. The Singular Value Decomposition (SVD) method is another famous recommendation model using the matrix factorization technique. PMF model is proposed by introducing Gaussian noise to observed value while SVD finds the matrix $\hat{R} = U\sum V^T$ of the given rank which minimizes the sum-squared distance to the target matrix R.
- *eALS* [14]. eALS is a matrix factorization method that weights unobserved data based on the popularity of products. This method does not require any auxiliary information.
- *MCA* [36]. MCA is a sequential pattern-based recommendation approach. It first mines a collection of sequential patterns and then recommends the remaining items after the occurrence of the prior items.

For methods with side information, we shall extract user/item-specific features and global features from Tuniu dataset. In particular, the features constructed in Section. 4.2 are taken as global features. We further add the membership of customers as one user-specific feature, and price, time costs as well as travel product types as three item-specific features.

- *RLFM* [1]. RLFM is a regression-based latent factor model for gaussian response. Here we fusing user-specific features, item-specific features and global features into the matrix factorization.
- *SVDFeature* [2]. SVDFeature is a model for feature-based collaborative filtering. Similar to RLFM, we use the same user-specific features, item-specific features and global features as model inputs.
- *BMFSI* [35]. BMFSI is a Bayesian matrix factorization method that utilizes auxiliary information. However, this model excludes global features. So we only use user-specific features

Table 2. Performance Comparisons (MAE and RMSE)

Method	50%		40%		30%		20%		10%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
PMF-MAI	**0.389**	**0.738**	**0.407**	**0.747**	**0.412**	**0.753**	**0.418**	**0.754**	**0.473**	**0.765**
UCF	0.509	0.828	0.538	0.866	0.595	0.883	0.684	0.992	0.785	1.110
ICF	0.684	1.097	0.733	1.144	0.795	1.181	0.876	1.233	0.994	1.298
PMF	0.453	0.788	0.468	0.795	0.479	0.801	0.508	0.810	0.548	0.847
SVD	0.496	0.817	0.519	0.857	0.545	0.859	0.606	0.901	0.722	0.910
eALS	0.432	0.765	0.446	0.769	0.459	0.782	0.467	0.789	0.512	0.816
RLFM	0.422	0.750	0.433	0.753	0.449	0.770	0.457	0.776	0.509	0.806
SVDFeature	0.428	0.762	0.437	0.768	0.458	0.779	0.463	0.786	0.517	0.812
BMFSI	0.451	0.771	0.459	0.774	0.461	0.789	0.482	0.793	0.531	0.821

and item-specific features as raw features to feed into BMFSI. For fair comparison, we further decompose global features into user (product) attributes to complement user-specific (item-specific) features, *e.g.,* user geolocation, user intended destination, departure and destination of travel product.

The dimension of latent factors, *i.e.,* K, is set to 10 by default for PMF, SVD, eALS, RLFM, SVDFeature and BMFSI and the proposed PMF-MAI. For our PMF-MAI, the sampling ratio γ is set to 0.3 and the weights in Eq. (15) are set equally by default, *i.e.,* $\lambda_{X1} = \lambda_{X2} = \lambda_{B1} = \lambda_U = \lambda_V = \lambda_{B2} = 0.05$. The impact of γ and other six weights will be discussed in Section 5.4. All experiments were done on the server with one quad-core E5-2650v2 processors and 128GB of main memory. PMF-MAI, PMF and MCA were implemented in Python by ourselves. We used the Mahout Java machine learning framework* to implement SVD, UCF, and ICF. For eALS, RLFM, SVDFeature and BMFSI, the source codes were available from Github.

5.2 The Overall Comparison

First of all, we present a performance comparison between PMF-MAI and baseline methods. To this end, we randomly split the elements in matrix $\mathbf{X}$ into training data and test data, and decrease training set ratio gradually from 50% to 10%. For each ratio of training data, we repeat the experiments for 10 times on different random splits and then report the average values of two performance metrics. The comparison results in terms of MAE and RMSE are shown in Table 2, where MCA is not reported since it directly generates recommender items. In Table 2, the best results are set to be bold. According to the results, there are several observations. First, our PMF-MAI method consistently gives rise to the lowest estimated errors in all splits, followed in decreasing order by RLFM, SVDFeature, eALS, BMFSI, PMF, SVD, UCF, and ICF. As the decrease in the number of training instances, PMF-MAI exhibits more superior performance. For example, PMF-MAI improves 1.6% on RMSE over RLFM with 50% training data, but it achieves 5.1% improvement on RMSE over RLFM with 10% training data. This clearly shows that PMF-MAI can better alleviate the data sparsity problem compared with other methods. Second, PMF-MAI, RLFM, SVDFeature and BMFSI perform better than PMF and SVD, demonstrating the effectiveness of incorporating auxiliary information. Furthermore, PMF-MAI, RLFM and SVDFeature beat BMFSI, the major difference among these methods is that BMFSI considers more user/item features, while other methods utilize

*https://mahout.apache.org/

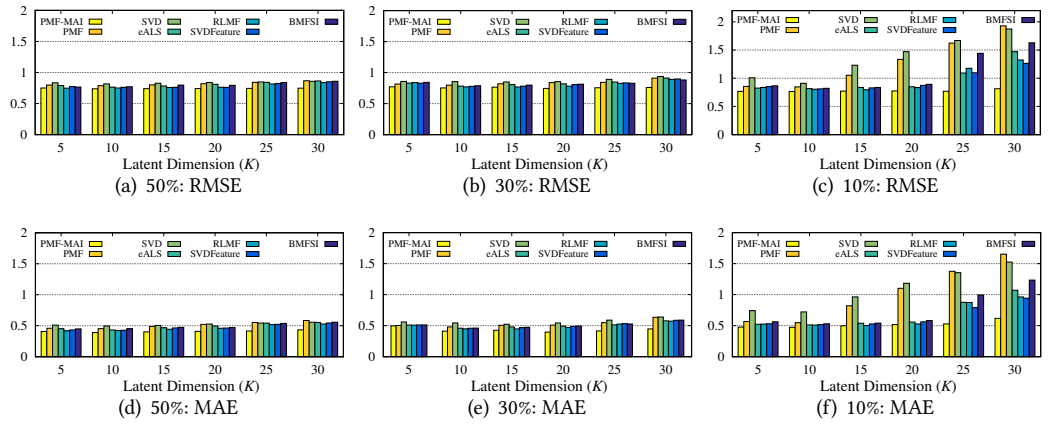

Fig. 4. The overall comparison results with the latent dimension K.

three kinds of features include global features, suggesting the benefit of jointing global features. Third, eALS achieves better performance than PMF and SVD because the use of the unobserved values. Unobserved values largely alleviate the data sparsity in the matrix factorization model and make the prediction more accurate. Last but not least, both UCF and ICF perform much worse than the models based on matrix factorization, which indicates that the neighborhood methods cannot work effectively on the extremely sparse user-product interaction matrix.

Fig. 4 shows the experimental results of various models with the varying parameter of the latent dimension size K. First of all, we can observe that PMF-MAI works well among these baselines in all cases. An intuitive explanation is that PMF-MAI leverages both auxiliary information and unobserved values to alleviate the data sparsity problem which implies the excellence of utilizing auxiliary information and unobserved values in learning latent factor of spare user-product interaction matrix for each user and item. Second, as the increase of K, PMF-MAI, RLFM, SVDFeature and BMFSI perform much more stable than PMF and SVD. This is largely owes to the auxiliary information integrated by these methods. Third, the performance improvement for all latent-based models is from K=5 to 10, and the prediction accuracy increases slowly and even decreases when the latent dimension further increases. This implies that the default setting $K = 10$ is fair to our method as well as its competitors and is reasonable for datasets with the varying ratio of the training set.

Fig. 5 shows the comparison results among ten methods in terms of Recall, F-measure, NDCG, and Coverage, respectively. We set the training set size as 50% and repeat the experiments 10 times on random splits. Then, the average values of each evaluation measure are adopted as the final results. We can observe several patterns from the results. First and foremost, our PMF-MAI significantly outperforms benchmark approaches indicated by all of the evaluation measures. The improvements of PMF-MAI achieved, on average, 13.8%, 13.1%, 2.7%, and 11.8% compared with the second-best performer RLFM in Recall, F-measure, NDCG, and Coverage, respectively. Second, PMF-MAI, RLFM, SVDFeature and BMFSI outperform PMF and SVD in terms of Recall, F-measure, NDCG and Coverage, again verifying the superiority of the frameworks for jointing auxiliary information and matrix factorization. Third, we observe that, by fusing only user-specific and item-specific features, BMFSI can only gain marginal improvements in terms of top-N recommendation performances. Compared with PMF without auxiliary information, BMFSI can only gain around 1.2%, 4.8%, and 0.2% improvement in terms of top-20 Recall, F-measure and NDCG respectively. We

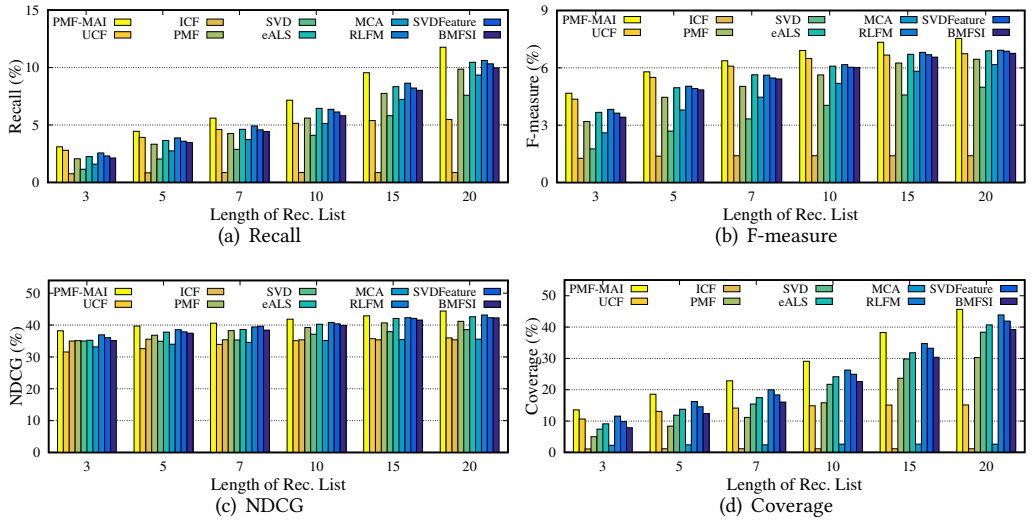

Fig. 5. Performance comparisons in terms of Recall, F-measure, NDCG, and Coverage. The training set size is set to 50%.

argue that user and product attributes are not fine-grained enough to discriminate user preferences. Quite differently, PMF-MAI, RLFM and SVDFeature leverage the constructed global features to better profile user interest preferences. Fourth, eALS is better than PMF and SVD. This is mainly because eALS imposes an item popularity-aware weighting strategy on unobserved values while PMF and SVD simply assign zeros to all missing values. Fifth, MCA dose not perform as well as PMF, especially in terms Coverage. Because MCA is frequency-based only, it is easy to filter out infrequent but significant items or patterns. That is, the sequential pattern-based approach is suitable for the relation discovery between frequent items in simple datasets. However, it may easily fail to model complex dependency in complex datasets for session-based recommendation. Sixth, UCF performs worser than MF-based methods on MAE and RMSE, but UCF obtains better overall accuracy than PMF and SVD as shown by Fig. 5(b). Closer inspection of Fig. 5(d) shows that the diversity of products within UCF's recommendation list is quite low. This observation suggests UCF tends to recommend popular products. Our PMF-MAI inherits the advantage of MF-based approaches that are able to recommend products having high diversity and yet improves the recommendation accuracy. Finally, both UCF and ICF appeared to be unaffected by the length of recommendation list, because every nearest neighbor has browsed few travel-products (*i.e.,* the data is extremely sparse) and thus the recommended products are limited.

5.3 Evaluation of Features Constructed by Auxiliary Information

The most striking characteristic of PMF-MAI is its ability of fusing a set of features and the matrix factorization. Six features in the case of e-travel scenario have been defined and used by PMF-MAI. We focus on evaluating these features, that is, to understand which features are more or less important to the prediction performance. We design two strategies for the evaluation. First, the standardized coefficients of the regression model can be used for evaluating the importance of every feature. Second, we shall fall back on some impurity measure commonly used in the classification model [5, 23] to evaluate the discriminative power of every feature. To this end, all of elements $x_{ij} \in \mathbf{X}$ are regarded as samples, and the integer value of x_{ij} is treated as the class label.

Table 3. Effect of implicit features on prediction accuracy

Type	Departure		Destination		Cost	
Feature	Geo. Sim.	Sem. Sim.	Geo. Sim.	Sem. Sim.	Price	Time
Fisher Score	0.0015	0.0008	0.0057	0.0020	0.0077	0.0054
Ranking	5	6	2	4	1	3
Std. Coefficients	0.020^{**}	0.008	0.029^{***}	0.011^{*}	0.047^{***}	0.039^{***}
Ranking	5	6	3	4	1	2

Note: $^{*}p < 0.05$; $^{**}p < 0.01$; $^{***}p < 0.001$.

As mentioned in Section 4.1, $x_{ij} \in [1, 26]$ and thus 26 classes are generated over all samples. Then, we select *Fisher score* [5] as the impurity measure for the evaluation of features. For each feature, the Fisher score is defined as

$$Fr = \frac{\sum_{c=1}^{26} n_c(\mu_i - \mu)^2}{\sum_{c=1}^{26} n_c \sigma_c^2}, \tag{28}$$

where n_c is the number of data samples in class c, μ_i is the average feature value in class c, σ_c is the standard deviation of the feature value in class c, and μ is the average feature value of all data samples. A higher Fisher score value indicates the feature has strong discriminative power.

Table 3 shows evaluation results of six features by using Fisher score and standardized coefficients. We can observe that the orders of features w.r.t. both metrics are almost consistent, except the order between the geographical similarity of destination and the time cost. However, the values of both metrics of the time cost correspond closely to those of the geographical similarity of destination. The top three features are price, geographical similarity of destination and time, where the p-value < 0.001 of their standardized coefficients indicates a very high significance level. Furthermore, Table 3 conveys the interesting information that tourists are more concerned with cost and destination but relatively care less about departure, when they are choosing travel products. As life quality rise, people increasingly prefer traveling individually and even by self-driving, rather than the traditional package tour. So, tourists are more inclined to purchase travel products on the destination, such as tickets, hotels, etc. Nevertheless, both financial and time costs are still the most important factors that tourists care about.

5.4 Effects of Parameters inside PMF-MAI

Here, we demonstrate the effects of parameters inside PMF-MAI, including the dimension of latent vector K, the usage of random sampling on unobserved data, and six weights of loss terms and regularization terms in Eq. (15).

5.4.1 Sampling Ratio. After utilizing random sampling technique, the computational complexity can be reduced to $O(KD(|\mathbf{X}| + \gamma(NM - |\mathbf{X}|)))$, where $\gamma \in [0, 1]$ is the sampling ratio. It is a tradeoff between the performance and the runtime to select the appropriate γ. The runtime per iteration is displayed in Fig. 6(a). Here, we use the same 10-dimensional latent features and the 50% training set ratio. It can be seen that the runtime is linearly increasing as the sampling ratio increases from 0 to 1 with 0.1 as the interval. On the other hand, Fig. 6(b)-6(f) depict the recommendation performances. We observe that the results of RMSE, Precision, Recall, NDCG and Coverage exhibit similar patterns. As sampling rate γ increases, the recommendation performances increase quickly at first. But when γ increases further, the recommendation performances increase slowly and even decreases. This phenomenon indicates that very high sampling rate cannot help to improve the recommendation

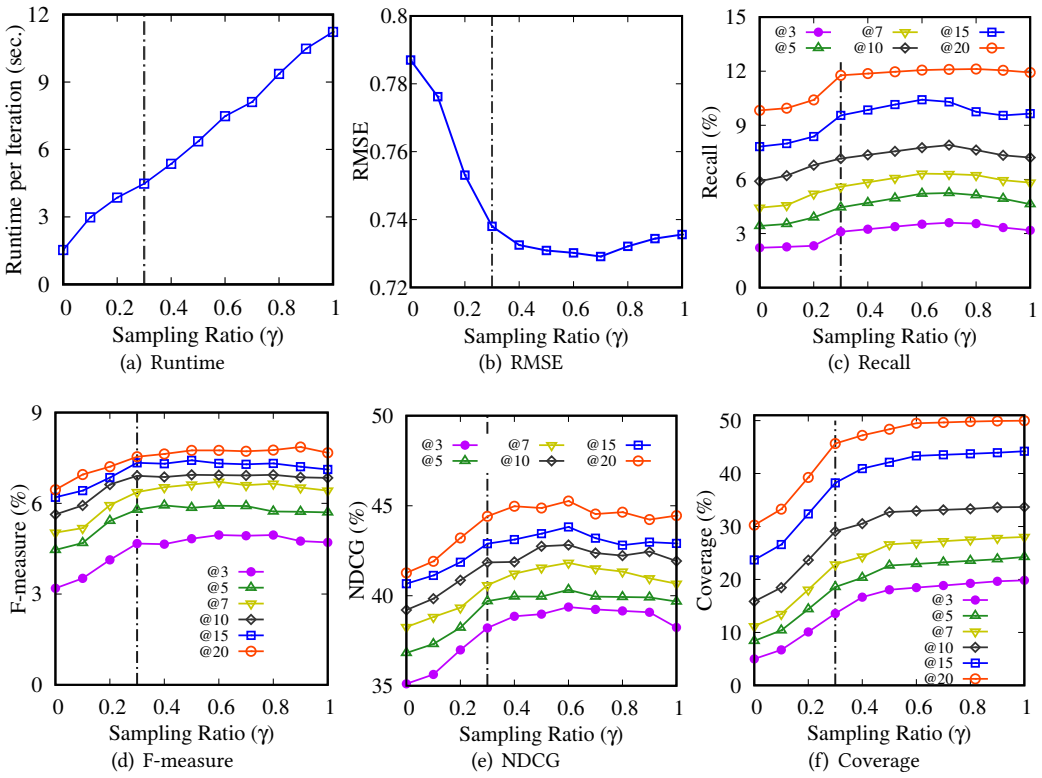

Fig. 6. Effect of sampling ratio on PMF-MAI performance.

performances. Consider the runtime and recommendation performances together, the appropriate γ is within [0.3,0.4]. This shows the default setting $\gamma = 0.3$, highlighted by a gray dash line in each figure, is reasonable. Compared with results learned on all unobserved data (*i.e.*, $\gamma = 1$), the runtime of each iteration is greatly reduces by 60% at the cost of increasing RMSE by 0.3%. Furthermore, it is noteworthy that the values at which the index curves intersect the y-axis represent the results of PMF-MAI without fusing unobserved values, *i.e.*, $\gamma = 0$. We can observe that Recall, F-measure, NDCG and Coverage of PMF-MAI without fusing unobserved values, on average, decrease by 20.8%, 20.7%, 6.6% and 47.7%. The major reason lies in extreme sparsity of user-product interaction matrix. Unlike other methods that assign zeros or weights to all unobserved values, PMF-MAI provides a new guide for incorporating unobserved values into matrix factorization model. The experimental results demonstrate the effectiveness of the way to incorporate unobserved values.

Remark: Our PMF-MAI directly employs the batch gradient descent as the optimization algorithm that calculates the error over all of samples. A limitation of this approach is that when the input matrix $\mathbf{X}$ grows larger, it requires the entire data in memory and the iterative optimization may become slower. Fortunately, there exist a number of variants of the gradient descent algorithm [10] such as the Stochastic Gradient Descent (SGD), the mini-batch gradient descent, and the Alternating Least Square (ALS) techniques, which are developed for accelerating the optimization on big data.

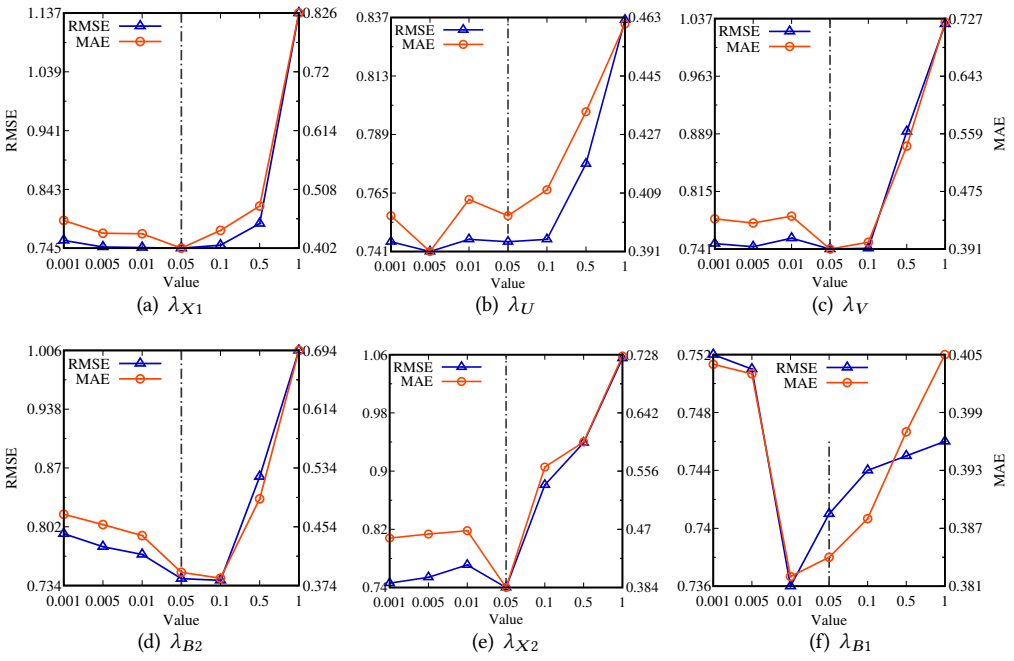

Fig. 7. Impact of six weights of loss terms and regularization terms.

5.4.2 Weights of Loss Terms and Regularization Terms. When introducing the PMF-MAI model above, we set the parameters $\lambda_{X1}, \lambda_U, \lambda_V, \lambda_{B2}, \lambda_{X2}, \lambda_{B1}$ to be equal, *i.e.*, λ_{X1}=λ_U=λ_V=λ_{B2}=λ_{X2}=λ_{B1}=0.05. We here verify its effectiveness by using the so-called traversal method. In this method, we alternately traverse the value of each parameter while keeping other parameters fixed. Fig. 7 exhibits the effects of six weights on the dataset of which the training set ratio is set to 50%. As can be seen, the default settings for six weights marked by the gray dash lines have achieved pretty good performance. As indicated by Figs 7(a), (c) and (e), $\lambda_{X1} = \lambda_V = \lambda_{X2} = 0.05$ is the best choice. In Figs 7(b) and (d), when setting λ_U and λ_{B2} to be default value, the performance is just slightly worse than the optimal performance. As shown in Fig 7(f), PMF-MAI has its best results for $\lambda_{B1} = 0.01$, which has a little difference from the default setting.

6 CONCLUSION

To tackle the travel-product recommendation problem, we presented a novel framework called PMF-MAI in this paper. Different from existing methods, PMF-MAI was generated by jointly considering user-item interaction matrix and multi-auxiliary information. Meanwhile, PMF-MAI could be viewed as a whole-data based learning framework that utilized unobserved click volumes as the calibration of probabilistic matrix factorization with linear regression. Experiments on a real-world online travel dataset have demonstrated PMF-MAI outperforms competitive baselines in terms of six evaluation measures, which is mainly attributed to fusing features constructed by auxiliary information. Up to now, the simple random sampling technique is employed by PMF-MAI to speed up its update on the large unobserved data. It will be important that future research integrates the learning of ranked unobserved values as a part of optimization objective in order to identify relevant unobserved instances for improving the performance. We also plan to seek more complex factorization models that can potentially lead to better latent representations.

REFERENCES

[1] Deepak Agarwal and Bee-Chung Chen. 2009. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 19–28.

[2] Tianqi Chen, Weinan Zhang, Qiuxia Lu, Kailong Chen, Zhao Zheng, and Yong Yu. 2012. SVDFeature: A toolkit for feature-based collaborative filtering. *Journal of Machine Learning Research* 13, Dec (2012), 3619–3622.

[3] Chen Cheng, Haiqin Yang, Irwin King, and Michael R Lyu. 2012. Fused matrix factorization with geographical and social influence in location-based social networks. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*. AAAI Press, 17–23.

[4] Robin Devooght, Nicolas Kourtellis, and Amin Mantrach. 2015. Dynamic matrix factorization with priors on unknown values. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 189–198.

[5] Richard O Duda, Peter E Hart, and David G Stork. 2012. *Pattern classification*. John Wiley and Sons.

[6] Asmaa Elbadrawy and George Karypis. 2015. User-specific feature-based similarity models for top-n recommendation of new items. *ACM Transactions on Intelligent Systems and Technology* 6, 3 (2015), 33.

[7] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized ranking metric embedding for next new POI recommendation. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*.

[8] Yong Ge, Qi Liu, Hui Xiong, Alexander Tuzhilin, and Jian Chen. 2011. Cost-aware travel tour recommendation. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 983–991.

[9] Yong Ge, Hui Xiong, Alexander Tuzhilin, and Qi Liu. 2014. Cost-aware collaborative filtering for travel tour recommendations. *ACM Transactions on Information Systems* 32, 1 (2014), 4.

[10] Rainer Gemulla, Erik Nijkamp, Peter J Haas, and Yannis Sismanis. 2011. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 69–77.

[11] Jiangning He, Hongyan Liu, and Hui Xiong. 2016. SocoTraveler: Travel-package recommendations leveraging social influence of different relationship types. *Information and Management* 53, 8 (2016), 934–950.

[12] Xiangnan He, Zhenkui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. 2018. NAIS: Neural Attentive Item Similarity Model for Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2018).

[13] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 549–558.

[14] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 549–558.

[15] Huiqi Hu, Yudian Zheng, Zhifeng Bao, Guoliang Li, Jianhua Feng, and Reynold Cheng. 2016. Crowdsourced POI labelling: Location-aware result inference and task assignment. In *32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, May 16-20, 2016*. 61–72.

[16] Jinlong Hu, Junjie Liang, Yuezhen Kuang, and Vasant Honavar. 2018. A user similarity-based top-n recommendation approach for mobile in-application advertising. *Expert Systems with Applications* (2018).

[17] C. Derrick Huang, Jahyun Goo, Kichan Nam, and Chul Woo Yoo. 2017. Smart tourism technologies in travel planning: The role of exploration and exploitation. *Information and Management* 54, 6 (2017).

[18] Shuhui Jiang, Xueming Qian, Tao Mei, and Yun Fu. 2016. Personalized travel sequence recommendation on multi-source big social media. *IEEE Transactions on Big Data* 2, 1 (2016), 43–56.

[19] Muhammad Usman Shahid Khan, Osman Khalid, Ying Huang, Rajiv Ranjan, Fan Zhang, Junwei Cao, Bharadwaj Veeravalli, Samee U Khan, Keqin Li, and Albert Y Zomaya. 2017. MacroServ: A route recommendation service for large-scale evacuations. *IEEE Transactions on Services Computing* 10, 4 (2017), 589–602.

[20] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[21] Rob Law, Shanshan Qi, and Dimitrios Buhalis. 2010. Progress in tourism management: A review of website evaluation in tourism research. *Tourism Management* 31, 3 (2010), 297–313.

[22] Huayu Li, Richang Hong, Defu Lian, Zhiang Wu, Meng Wang, and Yong Ge. 2016. A relaxed ranking-based factor model for recommender system from implicit feedback. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. AAAI Press, 1683–1689.

[23] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. 2017. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)* 50, 6 (2017), 94.

[24] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 831–840.

[25] Kwan Hui Lim, Jeffrey Chan, Shanika Karunasekera, and Christopher Leckie. 2017. Personalized itinerary recommendation with queuing time awareness. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 325–334.

[26] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. 2015. Personalized tour recommendation based on user interests and points of interest visit durations. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, Vol. 15. 1778–1784.

[27] Bin Liu, Hui Xiong, Spiros Papadimitriou, Yanjie Fu, and Zijun Yao. 2015. A general geographical probabilistic factor model for point of interest recommendation. *IEEE Transactions on Knowledge and Data Engineering* 27, 5 (2015), 1167–1179.

[28] Jie Liu, Bin Liu, Yanchi Liu, Huipeng Chen, Lina Feng, Hui Xiong, and Yalou Huang. 2018. Personalized air travel prediction: A multi-factor perspective. *ACM Transactions on Intelligent Systems and Technology* 9, 3 (2018), 30.

[29] Juntao Liu, Caihua Wu, Yi Xiong, and Wenyu Liu. 2014. List-wise probabilistic matrix factorization for recommendation. *Information Sciences* 278 (2014), 434–447.

[30] Qi Liu, Enhong Chen, Hui Xiong, Yong Ge, Zhongmou Li, and Xiang Wu. 2014. A cocktail approach for travel package recommendation. *IEEE Transactions on Knowledge and Data Engineering* 26, 2 (2014), 278–293.

[31] Qi Liu, Yong Ge, Zhongmou Li, Enhong Chen, and Hui Xiong. 2011. Personalized travel package recommendation. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*. IEEE Computer Society, 407–416.

[32] Pawel Matuszyk, João Vinagre, Myra Spiliopoulou, Alípio Mário Jorge, and João Gama. 2018. Forgetting techniques for stream-based matrix factorization in recommender systems. *Knowledge and Information Systems* 55, 2 (2018), 275–304.

[33] Sunho Park, Yong Deok Kim, and Seungjin Choi. 2013. Hierarchical bayesian matrix factorization with side information. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. 1593–1599.

[34] Chong Peng, Zhao Kang, Yunhong Hu, Jie Cheng, and Qiang Cheng. 2017. Nonnegative matrix factorization with integrated graph and feature learning. *ACM Transactions on Intelligent Systems and Technology* 8, 3 (2017), 42.

[35] Ian Porteous, Arthur U Asuncion, and Max Welling. 2010. Bayesian matrix factorization with side information and dirichlet process mixtures. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*.

[36] Cynthia Rudin, Benjamin Letham, Ansaf Salleb-Aouissi, Eugene Kogan, and David Madigan. 2011. Sequential event prediction with association rules. In *Proceedings of the 24th Annual Conference on Learning Theory*. 615–634.

[37] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic matrix factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*. Curran Associates Inc., 1257–1264.

[38] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2004. Collaborative filtering recommender systems. *ACM Transactions on Information Systems* 22, 1 (2004), 291–324.

[39] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, Lexing Xie, and Darius Braziunas. 2017. Low-rank linear cold-start recommendation from social data. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. 1502–1508.

[40] Ajit P Singh and Geoffrey J Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*. ACM, 650–658.

[41] Chang Tan, Qi Liu, Enhong Chen, Hui Xiong, and Xiang Wu. 2014. Object-oriented travel package recommendation. *ACM Transactions on Intelligent Systems and Technology* 5, 3 (2014), 43.

[42] Jiliang Tang, Xia Hu, Huiji Gao, and Huan Liu. 2013. Exploiting local and global social context for recommendation. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*. 2712–2718.

[43] Maksims Volkovs and Guang Wei Yu. 2015. Effective latent models for binary feedback in recommender systems. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 313–322.

[44] Mengting Wan, Di Wang, Matt Goldman, Matt Taddy, Justin Rao, Jie Liu, Dimitrios Lymberopoulos, and Julian McAuley. 2017. Modeling consumer preferences and price sensitivities from large-scale grocery shopping transaction logs. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1103–1112.

[45] Yisen Wang, Fangbing Liu, Shu-Tao Xia, and Jia Wu. 2017. Link sign prediction by variational bayesian probabilistic matrix factorization with student-t prior. *Information Sciences* 405 (2017), 175–189.

[46] Yu Ting Wen, Jinyoung Yeo, Wen Chih Peng, and Seung won Hwang. 2017. Efficient keyword-aware representative travel route recommendation. *IEEE Transactions on Knowledge and Data Engineering* 29, 8 (2017), 1639–1652.

[47] Dingqi Yang, Daqing Zhang, Vincent W Zheng, and Zhiyong Yu. 2015. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 1 (2015), 129–142.

[48] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential recommender system based on hierarchical attention networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*.

[49] Wayne Xin Zhao, Sui Li, Yulan He, Edward Y Chang, Ji-Rong Wen, and Xiaoming Li. 2015. Connecting social media to e-commerce: Cold-start product recommendation using microblogging information. *IEEE Transactions on Knowledge and Data Engineering* 28, 5 (2015), 1147–1159.

[50] Zhou Zhao, Hanqing Lu, Deng Cai, Xiaofei He, and Yueting Zhuang. 2016. User preference learning for online social recommendation. *IEEE Transactions on Knowledge and Data Engineering* 28, 9 (2016), 2522–2534.

[51] Vincent Wenchen Zheng, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative location and activity recommendations with GPS history data. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*. 1029–1038.

[52] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: Concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology* 5, 3 (2014), 38.

[53] Guixiang Zhu, Jie Cao, Changsheng Li, and Zhiang Wu. 2017. A recommendation engine for travel products based on topic sequential patterns. *Multimedia Tools and Applications* 76, 16 (2017), 17595–17612.

[54] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. 2018. Learning tree-based deep model for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2018, London, UK, August 19-23, 2018*. 1079–1088.