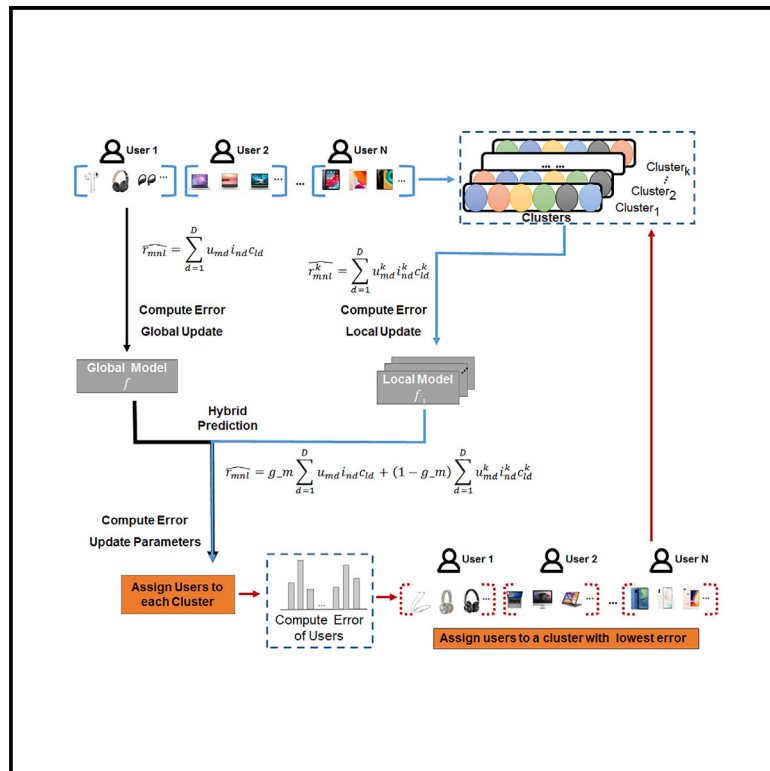


Patterns

Global and Local Tensor Factorization for Multi-criteria Recommender System

Graphical Abstract



Authors

Shuliang Wang, Jingting Yang, Zhengyu Chen, Hanning Yuan, Jing Geng, Zhen Hai

Correspondence

slwang2011@bit.edu.cn (S.W.), yhn6@bit.edu.cn (H.Y.), janegeng@bit.edu.cn (J.G.)

In Brief

In the multi-criteria recommendation system (often used in e-commerce), additional criterion-specific ratings can be used in addition to the existing user-item rating data. A new unified global and local tensor factorization method (GLTF) is proposed to obtain better recommendation results. This method can jointly learn a global predictive model and multiple local predictive models so that it is proficient in discovering the overall structure of the whole rating tensor and capturing diverse rating behaviors of users in individual subtensors.

Highlights

- A global and local tensor factorization is created for multi-criteria recommendation
- The method can learn a global predictive model and multiple local ones
- It discovers the structure of rating tensor and user-rating behaviors in subtensors
- It leverages user-item-criterion ratings for better recommendations in e-commerce



Article

Global and Local Tensor Factorization for Multi-criteria Recommender System

Shuliang Wang,^{1,2,*} Jingting Yang,³ Zhengyu Chen,⁴ Hanning Yuan,^{1,6,*} Jing Geng,^{1,2,*} and Zhen Hai⁵¹School of Computer Science, Beijing Institute of Technology, Beijing 100081, China²Institute of E-Government, Beijing Institute of Technology, Beijing 100081, China³College of Computer & Network Engineering, Shanxi Datong University, Datong 037009, China⁴College of Computer Science & Technology, Zhejiang University, Hangzhou 310007, China⁵Institute for Infocomm Research, 1 Fusionopolis Way, Singapore 138632, Singapore⁶Lead Contact*Correspondence: slwang2011@bit.edu.cn (S.W.), yhn6@bit.edu.cn (H.Y.), janegeng@bit.edu.cn (J.G.)<https://doi.org/10.1016/j.patter.2020.100023>

THE BIGGER PICTURE We propose a global and local tensor factorization method (GLTF) to solve the multi-criteria recommendation problem commonly experienced when e-commerce systems recommend products to users based on multiple different ratings. The method uses additional criterion-specific ratings in addition to existing user-item rating data for better recommendations. It can jointly learn a global predictive model and multiple local predictive models, not only by discovering the overall structure of the entire rating tensor but also by capturing diverse rating behaviors of users in individual subtensors. The GLTF can take advantage of the user's multi-criteria rating information to discover the user's behavior, predict the information and products that the user is interested in, and obtain more accurate recommendation results. In the future, we plan to apply the GLTF in a much larger dataset for evaluation and will improve the model to mitigate the bottleneck caused by the data sparsity problem.



Proof-of-Concept: Data science output has been formulated, implemented, and tested for one domain/problem

SUMMARY

In multi-criteria recommender systems, matrix factorization characterizes users and items via latent factor vectors inferred from user-item rating patterns. However, two-dimensional matrix factorization models may not be able to cope with the recommendation problem that involves additional criterion-specific rating data. This study introduces a tensor factorization method to handle three-dimensional user-item-criterion rating data. Moreover, we observe that using single global tensor factorization alone may not be sufficient to characterize diverse preferences among different groups of users, and a combined global and local tensor factorization method (GLTF) for multi-criteria recommendation is thus proposed. One key benefit of the GLTF is that it can leverage global user-item-criterion rating patterns while also exploiting local user-subset specific rating behaviors to jointly infer the latent factor representations for users, items, and specific item criteria. Experimental results, which used real-life data available to the public, demonstrated that the GLTF is superior to well-established baseline methods.

INTRODUCTION

This section introduces the study background and the previous work related to multi-criteria recommender systems. We then present preliminaries about matrix factorization techniques, and some related notations are presented in Table 1, which reveals the problem formulation.

Background

Recommender systems have become increasingly popular in a variety of online e-commerce websites and traveling portals. Traditional recommender systems typically operate on two-dimensional user-item rating data and seek to predict the preference or rating score of a user on a particular item (e.g., product).



Table 1. Meanings of Notations

Notation	Meaning
M	number of users
N	number of items
L	number of criteria
u_m	latent factor vector of user m
i_n	latent factor vector of item n
c_l	latent factor vector of criterion l
D	dimensionality of joint latent factor space
K	number of user subsets
u_m^k	latent factor vector of user m from subset k ($k: 1, \dots, K$)
i_n^k	latent factor vector of item n from subset k
c_l^k	latent factor vector of criterion l from subset k
r_{mnl}	observed rating of user m on the criterion l of item n
R	observed third-order user-item-criterion rating tensor

In recent years, various types of valuable information have become available in addition to the user-item ratings and have been investigated with recommender systems. In particular, different informative contexts, such as purchase intent, time, season, location, companion, and activity, have been leveraged to improve recommendation performance.^{1–9} Online unstructured textual reviews often contain users' opinions, attitudes, and preferences toward products or services and have been jointly exploited with the user-item rating data in various personalized recommendations.^{10–14}

By contrast, in this work, user ratings of multiple specific criteria of items, in addition to the overall user-item rating data, are considered to address the recommendation problem. On many leading e-commerce websites, online users are often allowed to rate their degree of satisfaction on multiple given criteria or aspects of products or services besides their overall ratings. Figure 1 shows an example of the multi-criteria rating system from TripAdvisor, where the user can not only give a single overall satisfaction rating of the hotel but also share their evaluation on each of three specific aspects related to the hotel, in this case value, service, and location. The performance of recommender systems can be greatly enhanced by exploiting a fine-grained multi-faceted representation of user preferences based on multi-criteria rating data.

A system that exploits multiple user ratings based on various criteria of items to support recommendations is commonly referred to as a multi-criteria recommender system (M-CRS). In the past few years, significant efforts have been made to deal with multi-criteria recommender systems. Existing approaches can be roughly grouped into three categories, namely the heuristic neighborhood-based approach, the aggregation-based approach, and the model-based approach. The heuristic neighborhood-based approaches first find a list of neighbors for a targeted user by using various multi-criteria similarity metrics to predict unknown ratings for the user based on the known ratings of the user's neighbors.^{15–19} Although the recommendation results are clearly explainable, the neighborhood-based approach tends to suffer from a sparsity of raw rating data, and it may not be scalable when dealing with large datasets. Assuming there is a certain relationship between overall item ratings and individual

criterion-specific ratings, the aggregation-based approach attempts to construct an aggregation function between them and then applies the function to aggregate the multiple criterion-specific ratings for prediction.^{17,20–24} By contrast, the model-based approach is primarily used to develop a predictive model by leveraging observed multi-criteria rating data and then use the model to predict a user's ratings of unknown items.^{25–30}

The different approaches have been proved to be robust for practical recommendation problems, and we have thus adopted the learning model-based approach to tackle the multi-criteria recommendation problem.

Previous studies have shown that matrix factorization methods are popular in recommender systems.^{31–33} Matrix factorization methods essentially characterize users and items via latent factor vectors learned from observed user-item rating data, such that the interactions between the users and items can be modeled as inner products of the two types of latent factor vectors. For multi-criteria recommendations, in addition to existing user-item ratings, multiple criterion-specific rating data are also available, and two-dimensional matrix factorization may not be able to cope with recommendations that involve additional multi-criteria rating patterns.

In this work, we propose to represent the user-item-criterion rating data as a third-order tensor and then introduce global tensor factorization (GTF) to deal with multi-criteria recommendations, where global means that the predictive model is learned from the whole set of rating data for all users. GTF extends classic matrix factorization and can factor the three-dimensional user-item-criterion tensor into low-dimensional representation. As a result, users, items, and criteria of items can be represented with low-dimensional vectors in a joint latent factor space. The resulting inner products of the user, item, and criterion vectors then capture the rating behaviors of the users.

The global model GTF predicts unknown ratings by learning from the whole set of observed user-item-criterion rating data, implicitly assuming that the distribution of the observed rating data is representative of the unknown data across all users. However, this assumption does not always hold true in reality because not all users behave in the same way. Recently, Beutel et al.³⁴ reported that a globally optimal model is typically not the best model to use for individual parts of the data. Although a global model is generally effective in estimating the overall structure as it relates to most or all users, it is often poor at detecting strong associations among individual small sets of closely related users.³⁵ If only a global model GTF is used for all users, the association among each subset of like-minded users would be ignored. This may result in an inaccurate similarity between a pair of users, especially those who have diverse preferences, which is a result of improper averaging, thereby reducing the personalized recommendation performance. In other words, the global model alone may not be sufficient to characterize the various preferences among different groups of users for recommendation.

To address this issue, we propose to partition the whole user-item-criterion rating tensor into multiple subtensors along the user dimension, whereby each subtensor collects the rating patterns of the subset of like-minded users. The GTF is then extended, and a local tensor factorization (LTF) method is developed that can learn multiple local predictive models from

Hilton Garden Inn South Padre Island

2,209 reviews #2 of 32 Hotels in South Padre Island
7010 Padre Blvd, South Padre Island, TX 78597-7726 00 1 855-937-0238 Visit hotel website

About

4.0 Very good
2,209 reviews

Location
Cleanliness
Service
Value

Certificate of Excellence

Property amenities

Free parking
Free High Speed Internet (WiFi)
Pool
Fitness Center with Gym / Workout Room
Restaurant
Beach
Bicycle rental
Business Center with Internet Access



andylloves2travel
London, United Kingdom

1 review
1 helpful vote

"Incredible London hotel"

Reviewed 3 weeks ago

Thank you for such a fantastic night's stay! All members of staff made the stay incredible memorable and enjoyable, including the customer relations manager on arrival, the housekeeping staff for an immaculately clean and imaginative bedroom, Augustin was such a wonderful host of a waiter for dinner and the smoked salmon and Dover Sole was simply scrumptious! A sensational hotel with amazing staff.

Stayed November 2016, travelled as a couple

Value Service Location

Helpful? 1 Thank andyloves2travel Report

Ask andyloves2travel about The Milestone Hotel

Figure 1. Multi-criteria Rating System from TripAdvisor

performance, such as contextual information (e.g., time, location, and companion, unstructured textual user reviews on items, and multiple user ratings on specific criteria of items).

Context-Aware Recommender Systems

Many definitions of context have been reported in previous studies, and common examples of this contextual information include time, location, companion, season, activity, and intent of a purchaser. Context has been recognized as an important factor for improving personalized recommendations. Palmisano et al.¹ exploited the contextual intents of purchases for predictive modeling of customers in personalized applications. To predict user ratings on items, Rendle et al.² proposed a context-aware factorization machine method that can tackle various types of context, such as mood of users, time, and location. Bhargava et al.³ leveraged the contextual information (i.e., who, what, when, and where) using a tensor factorization method for recommendation based on sparse user-generated data, while Yuan et al.⁴ ex-

exploited a similar context via a non-parametric Bayesian approach for recommendation and search for Twitter users. Wu et al.⁵ presented a contextual operating tensor method to handle a variety of interactive context data, such as companion, gender, age, occupation, and title. Ishanka and Yukawa⁶ chose two contextual parameters, i.e., emotion and user behavior, and implemented the travel destination recommendation system by using pre-filtering techniques and tensor factorization. Zheng⁷ proposed a simple but effective post-filtering algorithm to solve the problem of context-aware recommendation in mobile data.

Furthermore, Zheng and Jose⁸ proposed a novel context-aware recommendation mechanism in which user preferences are estimated by sequential predictions based on the sequence of context dimensions. Subsequently, Zheng et al.⁹ also tried to integrate context-awareness and multi-criteria decision making in the recommender systems by using the educational data as a case study. Their methods were able to capture the common semantic effects of context on users and items to improve recommendation performance.

Related Work

Recommender systems have become increasingly popular in recent years and have been widely used on a variety of e-commerce websites and traveling portals. In addition to well-known user-item rating data, modern recommender systems also need to handle other major types of data to improve recommendation

performance. The proposed LTF method takes diverse preferences among different groups of users into account and can recommend potential items to a targeted user by leveraging the preferences of their like-minded users in the same group.

Moreover, the proposed LTF has been found to be suitable for modeling diverse preferences of various subsets of users, especially when there are user subsets with diverse or even opposing preferences, while the proposed GTF still performs reasonably well at capturing overall rating patterns among the whole set of users. We have thus developed a new unified learning framework, named global and local tensor factorization (GLTF), which combines both GTF and LTF to deal with the multi-criteria recommendation problem. Our proposed GLTF method benefits from the advantages of both global and local tensor factorization, and it can not only jointly learn a global predictive model and multiple local predictive models but also simultaneously assign users to the local models.

Review-Aware Recommender Systems

In addition to user ratings of items, other major types of feedback that often come with item ratings include plain-text user reviews. User-generated review data are different from the aforementioned contextual information, as the textual reviews are typically unstructured. The review data often contain

Table 2. User-Item Rating Matrix Example

User	i_1	i_2	i_3	i_4	i_5
u_1	4	2	?	1	3
u_2	3	?	3	4	?
u_3	?	1	5	3	2
u_4	2	3	?	5	4
u_5	3	5	1	?	5

users' opinions, attitudes, and preferences toward products or services, and have been jointly exploited with the user-item ratings in various personalized recommendation systems.

McAuley and Leskovec¹⁰ proposed a hidden-factors- and topics-based method that combined latent rating dimensions with latent review topics to improve the product rating prediction problem. Bao et al.¹¹ developed a latent topic enhanced matrix factorization method to simultaneously leverage user ratings and textual reviews for recommendation. Zhang et al.¹² introduced collaborative multi-level embedded learning from text reviews for personalized rating prediction. Zheng et al.¹³ proposed a deep cooperative neural network approach to learn item properties and user behavior jointly from a review of text for recommendation. Cao et al.¹⁴ introduced a text-enhanced matrix factorization method to jointly exploit user rating and text data to improve cross-platform recommendations.

Multi-criteria Recommender Systems

The proposed study is similar to the aforementioned research because, in addition to well-known user-item ratings, various major types of available data have been leveraged to improve recommendations. However, instead of using contextual information or textual reviews, our proposed approach exploits multiple user ratings with respect to the specific criteria of items to improve recommendations. Employing multi-criteria ratings in recommender systems is not new, and existing techniques can be concisely grouped into three categories, namely heuristic neighborhood-based approaches, aggregation-based approaches, and model-based approaches. The heuristic neighborhood-based approach attempts to use various multi-criteria similarity metrics to collect the neighbors of a targeted user and then estimates unknown ratings based on the known ratings of those neighbors.¹⁵ Lakiotaki et al.¹⁶ calculated the distance between pairwise users using a multi-dimensional distance metric and employed a multi-criteria collaborative filtering method to identify the most preferred items for each given user. Liu et al.¹⁷ proposed a preference lattice based on user criteria preferences to predict the ratings for unknown items. Mikeli et al.¹⁸ estimated the overall distance between each pair of users using multi-criteria Euclidean distance and used a collaborative filtering technique to solve the recommendation problem. Syamala et al.¹⁹ proposed a novel technique to learn the criteria preferred by each user and also the criteria that made each item popular. This learning aided in finding similar user/item groups for recommending appropriate items to users. Although the recommendation results are often explainable, the neighborhood-based approaches tend to suffer from the sparsity of

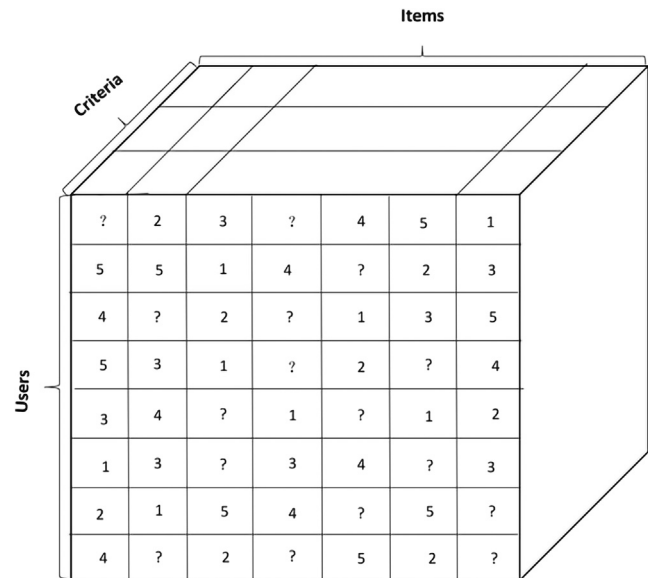


Figure 2. An Example of Third-Order User-Item-Criterion Rating Tensor

raw rating data and also may not be scalable when working with large datasets.

Assuming that there is a certain relation between overall user ratings and individual criterion ratings, the aggregation-based approaches primarily aim to build the mapping function to aggregate the multiple criterion-specific ratings for prediction. Lakiotaki et al.²⁰ proposed a utility additive method to aggregate the marginal users' preferences on the given criteria for recommendation. Jannach et al.²¹ proposed using a support vector regression to learn the relative importance of the individual criterion-specific ratings and then combined user- and item-based regression models using a weighted method to predict unknown ratings. Zheng²² proposed that the dependency among multiple criteria should be taken into account, and thus presented a criterion chain-based method to aggregate the multi-dimensional ratings for recommendation. Hamada et al.²³ proposed an aggregation function-based method that uses an adaptive genetic algorithm to efficiently incorporate the criteria ratings for improving the accuracy of the multi-criteria recommender system. In addition, Zheng²⁴ also proposed a utility-based multi-criteria recommendation algorithm that uses the vector of user expectations and evaluations to learn user expectations and establish utility functions.

On the contrary, the model-based approaches aim to learn a predictive model by leveraging observed multi-criteria rating data and then employing the model to estimate the ratings of a user on unknown items. Sahoo et al.²⁵ proposed a probabilistic mixture model-based algorithm to leverage the multiple component rating dependency structure for improving recommendation. Nilashi et al.²⁶ developed a recommendation method based on the adaptive neuro-fuzzy inference and self-organizing map-clustering models. Hamada et al.²⁷ presented a model that is based on the architecture and main features of fuzzy sets and systems to improve the prediction accuracy of

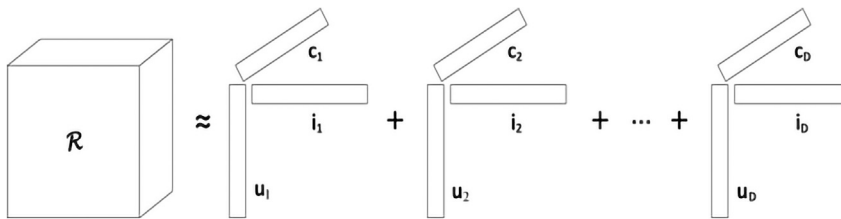


Figure 3. CP Tensor Decomposition Process

the recommender system. Li et al.²⁸ utilized a multi-linear singular value decomposition technique to explore the explicit and implicit relationships among user, item, and criteria for the recommendation task. Hassan and Hamada²⁹ proposed a neural network model trained using simulated annealing algorithms to improve the prediction accuracy of multi-criteria recommendation systems. Tallapally et al.³⁰ proposed extended stacked autoencoders (a deep neural network technique) to efficiently learn the relationship between each user’s criteria and overall rating.

The learning model-based approaches have been shown to be robust in practical recommendation systems. In this work, we employ the model-based technique, i.e., tensor factorization, to cope with the multi-criteria recommendation problem. High-order tensor, a generalization of matrix, is one of the powerful tools for modeling multi-faceted data, and various factorization techniques based on the tensor data have been developed for recommendation systems.³⁶

Rendle et al.³⁷ presented a ranking with tensor factorization algorithm to predict personalized tags for a user given an item. Karatzoglou et al.³⁸ used a high-order singular value decomposition (HOSVD) method to deal with contextual information in addition to the user-item data for context-aware recommendation problems. One limitation of HOSVD lies in that it primarily works for categorical context variables. Rendle et al.² extended HOSVD and proposed a factorization machine-based method to model various contextual data for context-aware rating prediction. Zheng et al.³⁹ proposed to represent the user-location-activity relations via the third-order tensor and developed a regularized tensor and matrix decomposition method for location and

Bhargava et al.³ developed a straightforward tensor factorization method to tackle context-aware collaborative recommendation, while Yao et al.⁴¹ presented a social regularization-based tensor factorization method for point-of-interest recommendation problem.

On the contrary, our proposed GLTF method is different from the aforementioned factorization techniques. On one hand, our method leverages multiple criterion-specific ratings in addition to user-item data and is proposed to deal with the multi-criteria recommendation problem. We not only aim to predict overall ratings of users on unknown items but also aim to deal with fine-grained criterion-specific rating prediction for recommendation. On the other hand, the proposed method is not only able to build a global predictive factorization model by discovering overall structure of the user-item-criterion tensor data but is also able to learn multiple local predictive models via factoring user-subset specific subtensors; moreover, both global and local factorization models are jointly employed to predict unknown ratings for more accurate recommendation.

We note that discovering the local structure of observed rating data is helpful for improving recommender systems.^{42,43} Assuming that the rating matrix is locally of low rank, Lee et al.⁴⁴ proposed a local low-rank matrix approximation method for rank-based recommendation.⁴⁵ Co-clustering for users and items has been also shown to be effective for improving collaborative recommendation tasks.^{46–48} To the best of our knowledge, almost all existing approaches to mining local behavioral patterns for the purposes of providing recommendations were developed to handle two-dimensional user-item data and thus

Algorithm 1. Global Tensor Factorization Method

```

1: Input:
2:    $T$ : A set of known user-item-criterion ratings  $r_{mnl}$ ;
3:    $T$ : Maximum number of iterations;
4:    $\alpha_G, \beta_G$ : Hyper-parameters;
5:    $\lambda$ : Learning rate.
6: Initialization:
7:   Initialize  $u_m \in \mathbb{R}^D$ ,  $i_n \in \mathbb{R}^D$ , and  $c_l \in \mathbb{R}^D$  randomly for each tuple  $(m, n, l)$ .
8: For  $t = 1, \dots, T$  do
9:   Randomly shuffle examples in the known training set  $T$ .
10:  For each example  $(m, n, l)$  in  $T$  do
11:    Make a prediction  $\widehat{r}_{mnl}$  via Equation 4;
12:    Compute predictive error  $e_{mnl}$ ;
13:    Update parameters  $u_{md}$ ,  $i_{nd}$ , and  $c_{ld}$  via Equation 6.
14:  End for
15: End for

```

Algorithm 2. Local Tensor Factorization Method

```

1: Input:
2:    $K$ : Number of user clusters;
3:    $T^k$ : A subset of known user-item-criterion ratings  $r_{mnl}$  for user cluster  $k$ ;
4:    $T$ : Maximum number of iterations;
5:    $\alpha_L, \beta_L$ : Hyper-parameters;
6:    $\lambda$ : Learning rate.
7: Initialization:
8:   Initialize  $u_m^k \in \mathbb{R}^D$ ,  $i_n^k \in \mathbb{R}^D$ , and  $c_l^k \in \mathbb{R}^D$  randomly for each tuple  $(m, n, l)$ .
9: While Significant change in the assignments is detected do
10:  For  $t = 1, \dots, T$  do
11:    For  $k = 1, \dots, K$  do
12:      Randomly shuffle examples in the known training set  $T^k$ .
13:      For each example  $(m, n, l)$  in  $T^k$  do
14:        Make a prediction  $\widehat{r}_{mnl}^k$  via Equation 7;
15:        Compute predictive error  $e_{mnl}^k$ ;
16:        Update parameters  $u_{md}^k$ ,  $i_{nd}^k$ , and  $c_{ld}^k$  via Equation 9.
17:      End for
18:    End for
19:  End for
20:  For  $m = 1, \dots, M$  do
21:    Assign user  $m$  to each of  $K$  clusters;
22:    Compute respective predictive errors based on updated parameters;
23:    Identify the cluster  $k$  for user  $m$ , where the lowest error is achieved.
24:  End for
25: End while

```

may not be able to handle issues that involve additional criterion-dependent rating data. One key advantage of our proposed method is that it can learn both global and local predictive models by jointly capturing overall rating structures and those specific to user subsets of third-order user-item-criterion data.

Preliminaries

Previous studies have indicated that matrix factorization and their variants are the dominant techniques used in modern recommender systems.^{32,33,45}

Basically, in recommendation, matrix factorization models deal with two-dimensional preference relations between users and items. Table 2 shows an example of pairwise user-item ratings whereby each user is allowed to flexibly rate a given item on a 5-point rating scale from 1 to 5, while “?” refers to unknown ratings.

Based on known ratings data, the model then projects the users and items into a joint latent factor space, such that the user-item preferences can be modeled as the inner products of the latent factors in the space. Formally, let u_m be a latent factor representation derived from the matrix factorization model for user m , and i_n is the latent representation for item n . The preference rating \widehat{r}_{mn} by user m of item n can then be estimated according to Equation 1,

$$\widehat{r}_{mn} = u_m^T i_n. \quad (\text{Equation 1})$$

Clearly, the key challenge in the recommendation system is how to derive the representations of users and items in the joint

latent factor space. To accomplish this, the regularized squared loss on the set of observed user-item rating data is minimized using Equation 2,

$$\begin{aligned} \text{loss} = \min_{\{u^*, i^*\}} \sum_{(m,n) \in O} \frac{1}{2} (r_{mn} - \widehat{r}_{mn})^2 + \frac{\beta}{2} (\|u_m\|_2^2 + \|i_n\|_2^2) \\ + \alpha (\|u_m\|_1 + \|i_n\|_1), \end{aligned} \quad (\text{Equation 2})$$

where O refers to the set of user-item pairs (m, n) for which the ratings r_{mn} are known. The first term is the squared prediction error, and both the second and third terms are L2-norm and L1-norm regularizes that control model complexity, where α and β are hyper-parameters. The optimization problem in Equation 2 can be solved using the classic stochastic gradient descent method, which iteratively updates the latent factor vectors of users and items.⁴⁹ Once the optimization process is done, Equation 1 can then be used to straightforwardly predict the ratings of a given user of unknown items.

Problem Formulation

Generally, multi-criteria recommender systems (MCRSs) refer to systems that leverage multiple user ratings of various item criteria in addition to overall user-item ratings to support recommendations. Following Adomavicius and Kwon,¹⁵ the MCRS can be formulated as follows:

$$U \times I \times C \rightarrow R_0 \times R_1 \times \dots \times R_{L-1},$$

where U, I , and C on the left side are the sets of users, items, and item criteria, respectively, while on the right, R_0 represents the

Algorithm 3. Global and Local Tensor Factorization Method

```

1: Input:
2:    $K$ : Number of user clusters;
3:    $T^k$ : A subset of known user-item-criterion ratings  $r_{mnl}$  for user cluster  $k$ ;
4:    $T$ : Maximum number of iterations;
5:    $\alpha_G, \beta_G, \alpha_L, \beta_L$ : Hyper-parameters;
6:    $\lambda$ : Learning rate;
7:    $g_m$ : Personalized weight initialized as 0.5 for each user  $m$ .
8: Initialization:
9:   Randomly initialize global latent vectors  $u_m \in \mathbb{R}^D$ ,  $i_n \in \mathbb{R}^D$ , and  $c_l \in \mathbb{R}^D$ , and local latent vectors  $u_m^k \in \mathbb{R}^D$ ,  $i_n^k \in \mathbb{R}^D$ , and  $c_l^k \in \mathbb{R}^D$  for each tuple  $(m, n, l)$ .
10: While Significant change in the assignments is detected do
11:   For  $t = 1, \dots, T$  do
12:     For  $k = 1, \dots, K$  do
13:       Randomly shuffle examples in the training set  $T^k$ .
14:       For each example  $(m, n, l)$  in  $T^k$  do
15:         Make a prediction  $\widehat{r}_{mnl}$  via Equation 10;
16:         Compute predictive error  $e_{mnl}$ ;
17:         Update parameters  $u_{md}$ ,  $i_{nd}$ ,  $c_{ld}$ ,  $u_{md}^k$ ,  $i_{nd}^k$ , and  $c_{ld}^k$  via Equation 12;
18:         Update parameter  $g_m$  via Equation 13.
19:       End for
20:     End for
21:   End for
22:   For  $m = 1, \dots, M$  do
23:     Assign user  $m$  to each of  $K$  clusters;
24:     Compute respective predictive errors based on updated parameters;
25:     Identify the cluster  $k$  for user  $m$ , where the lowest error is achieved;
26:     Update personalized weight parameter  $g_m$ .
27:   End for
28: End while

```

overall ratings of the items by users, and R_1, R_{L-1} represent user ratings of individual item criteria (L is the number of criteria). Note that the overall rating information is treated as a special type of criterion rating in the formulation.

Given observed user-item-criterion rating data, a multi-criteria predictive model must first be built by fitting the observed data, and the model is then applied to predict the overall ratings as well as multiple criterion-specific ratings that a user would give to unknown items.

Naturally, we introduce a third-order tensor, a generalization of matrix, to represent the three-dimensional user-item-criterion rating data. Figure 2 shows an example, in this case a toy, which uses a third-order user-item-criterion rating tensor whereby each user rates various criteria of the item on a rating scale from 1 to 5, with the question mark indicating an unknown rating. Next, several factorization methods were developed based on the tensor data or, more specifically, GLTF methods, to obtain the multi-criteria recommendations.

Clearly, two-dimensional matrix factorization techniques may not be able to provide recommendations that involve multiple criterion-specific ratings in addition to user-item data. Thus, the classic matrix factorization model was expanded, and tensor factorization methods were used to learn predictive models based on the three-dimensional user-item-criterion rating data.

RESULTS

We introduced a GTF method that is proficient in modeling overall rating behaviors of the whole set of users. We then employed an LTF method to characterize the diverse preferences among different groups of users. To take advantage of both methods, we developed a unified learning framework (i.e., GLTF method) to address the issues associated with providing multi-criteria recommendations. Our new GLTF method can jointly learn one global and multiple local predictive models while it simultaneously tackles the assignment of users to the local models. Furthermore, we validate various proposed methods and also compare them with existing approaches for rating prediction, evaluating statistical significance of the differences with p values. The effect of initialization methods is then studied for clustering users, and the impacts of varying numbers of user clusters and latent factors are evaluated respectively. We analyze the interplay between global and local models for prediction.

Global Tensor Factorization

We generally described classic matrix factorization in the previous section Preliminaries, and proposed a GTF method, where global means that a single predictive model was used in estimations for all the users.

For third-order user-item-criterion tensor data, the objective of tensor factorization is to map the users, items, and criteria in a

Table 3. MAE of Various Methods on Three Different Datasets ($p = 0.95$)

Method	TripAdvisor	Yahoo!Movie	RateBeer
GTF	0.6878 ± 0.0326	0.6217 ± 0.0286	0.6471 ± 0.0125
LTF	0.6724 ± 0.0085	0.5966 ± 0.0179	0.6206 ± 0.0059
GLTF ₀	0.6775 ± 0.0183	0.5992 ± 0.0301	0.6387 ± 0.0057
GLTF _f	0.6425 ± 0.0123	0.5509 ± 0.0159	0.5800 ± 0.0109
GLTF	0.6178 ± 0.0163	0.5076 ± 0.0155	0.5747 ± 0.0089
AFBM	0.8638 ± 0.0117	0.6509 ± 0.054	0.7026 ± 0.0062
CC	0.8258 ± 0.0935	0.6177 ± 0.0099	0.6460 ± 0.0077

The lowest MAE of each dataset is highlighted in bold type.

joint latent factor space, such that the preferences of the users with respect to specific item criteria can be formulated as inner products of corresponding latent factor vectors in the space. Perhaps one of the most popular tensor factorization paradigms is CANDECOMP/PARAFAC (CP), possibly due to its key advantage of linear time complexity.²⁹ Hence, we employed CP to factor the rating tensor data.

In the GTF model, given a user-item-criterion rating tensor $R \in \mathbb{R}^{M \times N \times L}$, CP can decompose the tensor into a sum of rank-1 tensors across the entire set of users with Equation 3,

$$R \approx \sum_{d=1}^D u_d \circ i_d \circ c_d, \quad (\text{Equation 3})$$

where $u_d \in \mathbb{R}^M$, $i_d \in \mathbb{R}^N$, $c_d \in \mathbb{R}^L$, D is the dimensionality of the joint latent space, and the symbol \circ means the vector outer product. Figure 3 shows the CP decomposition of the third-order user-item-criterion tensor.

Then, according to CP decomposition, the preference rating by user m of criterion l of item n can be estimated using Equation 4,

$$\widehat{r}_{mnl} = \sum_{d=1}^D u_{md} i_{nd} c_{ld}, \quad (\text{Equation 4})$$

where $u_m \in \mathbb{R}^D$, $i_n \in \mathbb{R}^D$, and $c_l \in \mathbb{R}^D$ are the latent factor representations of user m , item n , and criterion l , respectively. The resulting inner product of the latent vectors describes the interactive relationship among the given tuple of user, item, and criterion. Once the latent factor representations are learned, the rating prediction can then be accomplished straightforwardly via Equation 4. To address this, we minimized the following regularized squared loss on the observed set of user-item-criterion rating data:

$$\begin{aligned} \text{loss} = & \min_{\{u, i, c\}} \sum_{(m,n,l) \in T} \frac{1}{2} (r_{mnl} - \widehat{r}_{mnl})^2 + \frac{\beta_G}{2} (\|u_m\|_2^2 + \|i_n\|_2^2 + \|c_l\|_2^2) \\ & + \alpha_G (\|u_m\|_1 + \|i_n\|_1 + \|c_l\|_1), \end{aligned} \quad (\text{Equation 5})$$

where T is the set of user-item-criterion tuples (m, n, l) for which r_{mnl} is known (i.e., training set) and \widehat{r}_{mnl} is the predicted rating by user m of criterion l of item n using Equation 4. The first term in Equation 5 is the squared error between the observed and predicted ratings. The second and third terms are L2-norm and L1-norm regularizers, where β_G and α_G are hyper-parameters, respectively.

Following Koren et al.,³² we employ stochastic gradient descent (SGD) to optimize the loss from the GTF method. SGD loops through the entire observed user-item-criterion ratings in the training set. For each given training example (m, n, l) , the system first makes a prediction \widehat{r}_{mnl} and then calculates the predictive error as

$$e_{mnl} = r_{mnl} - \widehat{r}_{mnl}.$$

Using SGD, the parameters are then updated by a magnitude proportional to the learning rate λ in opposition to the gradient, yielding Equation 6,

$$\begin{cases} u_{md} = u_{md} - \lambda(\alpha_G + \beta_G u_{md} - e_{mnl} i_{nd} c_{ld}) \\ i_{nd} = i_{nd} - \lambda(\alpha_G + \beta_G i_{nd} - e_{mnl} u_{md} c_{ld}) \\ c_{ld} = c_{ld} - \lambda(\alpha_G + \beta_G c_{ld} - e_{mnl} u_{md} i_{nd}) \end{cases}. \quad (\text{Equation 6})$$

The pseudocodes of the proposed GTF method are summarized in Algorithm 1. After the system completes the training process, the learned global model can be straightforwardly employed to predict the overall ratings that a user gives to unknown items, as well as their specific ratings of item criteria.

Local Tensor Factorization

The GTF model is generally good at discovering the overall structure that relates to most or all users. However, the global model may not be able to detect the strong associations among individual subsets of closely related users.³⁵ In other words, if using only a single global model for all users, the similarity between a pair of users who typically have different preferences would tend to be inaccurately represented by some average value. Thus, it may not be sufficient to build a global factorization model alone if the objective is to capture diversified preferences of all the users, especially when there are user subsets with different or even opposite preferences.

To address this critical issue, we developed a new LTF method for multi-criteria recommendation systems. In particular, LTF first assigns each given user to a subset that consists of their like-minded users and then partitions the entire third-order user-item-criterion tensor into multiple subtensors according to the user subsets. Next, CP decomposition is employed to factor individual subtensors and learn multiple local user-subset specific predictive models. One key benefit of LTF is that it primarily recommends a user for the items enjoyed only by their associated subset of like-minded users.

Formally, let subset k be the cluster that contains given user m , and R^k be the corresponding user-item-criterion subtensor for the subset. The user-subset specific prediction for the rating by user m of criterion l of item n on the subtensor can then be made:

$$\widehat{r}_{mnl}^k = \sum_{d=1}^D u_{md}^k i_{nd}^k c_{ld}^k, \quad (\text{Equation 7})$$

where u_m^k , i_n^k , and c_l^k are user-subset specific latent factor representations for given user m , item n , and criterion l on the local subtensor R^k , respectively. Next, to learn the local latent factor vectors of users, items, and criteria for a given particular user-subset, we propose to minimize the following regularized squared loss of the LTF method:

Table 4. Dataset Description

Dataset	Users	Items	Records	Sparsity	Criteria
TripAdvisor ⁵⁰	6,134	1,763	23,066	99.79%	value, location, service, and overall
Yahoo!Movie ²⁵	1,827	1,479	50,673	98.13%	story, acting, direction, visual effects, and overall
RateBeer ⁵¹	3,630	4,896	48,605	99.73%	appearance, aroma, palate, taste, and overall

$$\text{loss} = \min_{\{U_m^k, I_n^k, C_l^k\}} \sum_{k=1}^K \sum_{(m,n,l) \in T^k} \frac{1}{2} (r_{mnl} - \widehat{r}_{mnl}^k)^2 + \frac{\beta_L}{2} (\|U_m^k\|_2^2 + \|I_n^k\|_2^2 + \|C_l^k\|_2^2) + \alpha_L (\|U_m^k\|_1 + \|I_n^k\|_1 + \|C_l^k\|_1), \quad (\text{Equation 8})$$

where K is the number of user subsets, T^k is a user-subset specific to user-item-criterion tuples (m, n, l) for which the ratings are known, and both β_L and α_L are hyper-parameters. Given a user-subset k and a known training example (m, n, l) , the first term represents the squared prediction error (i.e., $e_{mnl}^k = r_{mnl} - \widehat{r}_{mnl}^k$), and both second and third terms are L2-norm and L1-norm regularizers, respectively.

SGD is employed to optimize the loss function of the local factorization model, and the user-subset specific parameters are updated using Equation 9.

$$\begin{cases} U_{md}^k = U_{md}^k - \lambda(\alpha_L + \beta_L U_{md}^k - e_{mnl}^k I_{nd}^k C_{ld}^k) \\ I_{nd}^k = I_{nd}^k - \lambda(\alpha_L + \beta_L I_{nd}^k - e_{mnl}^k U_{md}^k C_{ld}^k) \\ C_{ld}^k = C_{ld}^k - \lambda(\alpha_L + \beta_L C_{ld}^k - e_{mnl}^k U_{md}^k I_{nd}^k) \end{cases} \quad (\text{Equation 9})$$

To find clusters of like-minded users, LTF adopts a heuristic approach that can jointly tackle the assignment of users to individual subsets and learn the predictive models that achieve lower predictive error.

Specifically, all users are initially partitioned into K clusters randomly or by using an existing clustering method. During training, prediction errors are obtained for the assignment of each user to different clusters. The assignment of the user is then adjusted to an appropriate cluster for which the lowest prediction error is achieved. The process is performed iteratively until no significant change in the assignments is detected, where

significant change refers to the number of users switching clusters is more than 1% of total number of given users. Algorithm 2 summarizes the main steps of the proposed LTF method.

Global and Local Tensor Factorization

The GTF method can discover the overall behavioral patterns from the whole set of user-item-criterion rating data, while the LTF method takes diverse preferences into account among different subsets of like-minded users and is thus adept at mining local interactive behaviors for personalized recommendation. In fact, the local factorization method optimizes the loss for individual user subsets; thus, the users with more observed ratings in a subset are often considered to be more important than users with fewer ratings. As a result, the learned local predictive models tend to be biased toward relatively popular users within individual subsets.

To take advantage of the above two factorization methods, we then developed a new unified learning framework, i.e., a GLTF method that provides multi-criteria recommendations. Notably, GLTF can leverage global user-item-criterion interactive patterns while also exploiting local user-subset specific preference behaviors to derive latent factor representations for users, items, and specific item criteria.

When using GLTF, the whole set of users must first be partitioned into various subsets, and respective subtensors must be obtained by dividing given third-order user-item-criterion tensor data according to the user subsets. The local latent factors for users, items, and criteria are derived based on the subtensors, and generate the global latent factor representations based on the whole rating tensor. Then, to estimate the rating \widehat{r}_{mnl} that a user m gives to criterion l of an item n , the global model and corresponding local predictive models for user-subset k are used together as follows:

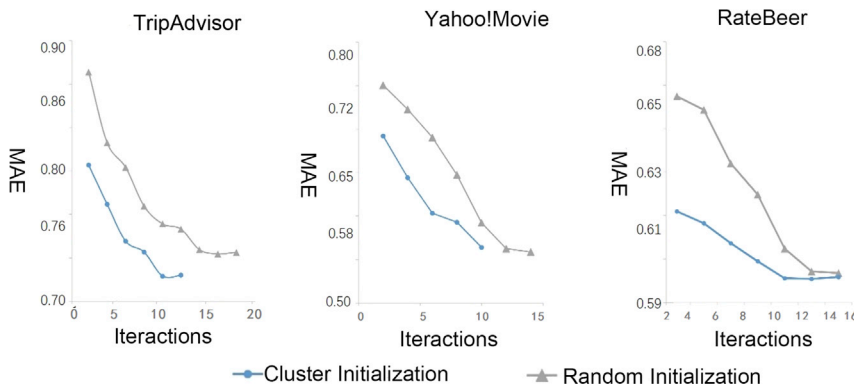


Figure 4. Comparison between Two Initialization Methods for User Subsets on Three Different Datasets

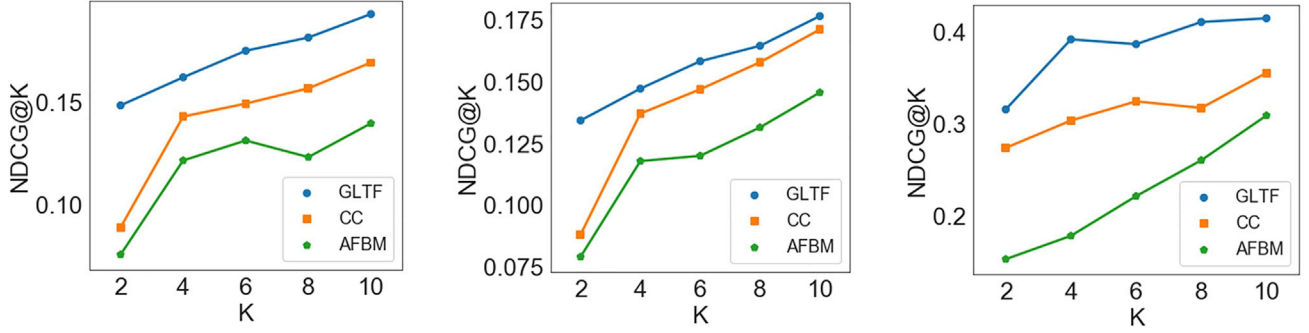


Figure 5. Top-K Recommendation in Terms of NDCG@K on Three Different Datasets

$$\widehat{r}_{mnl} = g_m \sum_{d=1}^D u_{md} i_{nd} c_{ld} + (1 - g_m) \sum_{d=1}^D u_{md}^k i_{nd}^k c_{ld}^k, \quad (\text{Equation 10})$$

where u_m , i_n , and c_l refer to global latent factor representations of users, items, and criteria, respectively, while u_m^k , i_n^k , and c_l^k correspond to their local latent factor representations. g_m is a personalized hyper-parameter that tunes the interplay between global and local predictive models. When g_m is equal to 1, GLTF would be reduced to GTF for rating prediction, and it would then be reduced to LTF when g_m is equal to 0.

Next, based on multiple subsets of known user-item-criterion rating data, the following combined squared loss of the GLTF method is minimized to jointly learn the optimal global and local latent factor vectors for users, items, and criteria of the items, as expressed in Equation 11,

$$\begin{aligned} \text{loss} = & \min_{\{u^*, i^*, c^*\}, \{u^{k*}, i^{k*}, c^{k*}\}} \sum_{k=1}^K \sum_{(m,n,l) \in T^k} \frac{1}{2} (r_{mnl} - \widehat{r}_{mnl})^2 \\ & + \frac{\beta_G}{2} (\|u_m\|_2^2 + \|i_n\|_2^2 + \|c_l\|_2^2) + \alpha_G (\|u_m\|_1 + \|i_n\|_1 + \|c_l\|_1) \\ & + \frac{\beta_L}{2} (\|u_m^k\|_2^2 + \|i_n^k\|_2^2 + \|c_l^k\|_2^2) + \alpha_L (\|u_m^k\|_1 + \|i_n^k\|_1 + \|c_l^k\|_1). \end{aligned} \quad (\text{Equation 11})$$

The first term is the squared prediction error, both second and third terms are global L2-norm and L1-norm regularizers, and the last two terms are local L2-norm and L1-norm regularizers. β_G , α_G , β_L , and α_L are respective hyper-parameters. The optimization problem in Equation 11 can be solved using the SGD method.

$$\begin{cases} u_{md} = u_{md} - \lambda(\alpha_G + \beta_G u_{md} - e_{mnl} g_m i_{nd} c_{ld}) \\ i_{nd} = i_{nd} - \lambda(\alpha_G + \beta_G i_{nd} - e_{mnl} g_m u_{md} c_{ld}) \\ c_{ld} = c_{ld} - \lambda(\alpha_G + \beta_G c_{ld} - e_{mnl} g_m u_{md} i_{nd}) \\ u_{md}^k = u_{md}^k - \lambda\alpha_L + \beta_L u_{md}^k - e_{mnl} (1 - g_m) i_{nd}^k c_{ld}^k \\ i_{nd}^k = i_{nd}^k - \lambda\alpha_L + \beta_L i_{nd}^k - e_{mnl} (1 - g_m) u_{md}^k c_{ld}^k \\ c_{ld}^k = c_{ld}^k - \lambda\alpha_L + \beta_L c_{ld}^k - e_{mnl} (1 - g_m) u_{md}^k i_{nd}^k \end{cases} \quad (\text{Equation 12})$$

The global predictive model and local predictive models are combined with personalized weights g_m , which is updated automatically. To compute the personalized weight g_m , we

minimize the squared loss of Equation 11 for user m , which comes from subset k , over all items n and criteria l . By setting the derivative of the squared loss to 0, we get Equation 13,

$$g_m = \frac{\sum_{(m,n,l) \in T^k} r_{mnl} - \sum_{d=1}^D u_{md}^k i_{nd}^k c_{ld}^k}{\sum_{d=1}^D u_{md} i_{nd} c_{ld} - \sum_{d=1}^D u_{md}^k i_{nd}^k c_{ld}^k}, \quad (\text{Equation 13})$$

where S is the total number of user-item-criterion ratings given by user m . After learning the global model and local models, GLTF updates the personalized weight g_m for each user with Equation 13. GLTF assigns every user m to each possible subset. In each subset, the weight g_m and training error are calculated. Thus, user m would be assigned to the subset with the smallest training error. Note that if there is no subset for which the training error is smaller, user m remains in the same subset. The process is performed iteratively until it converges. The main steps of GLTF are summarized by Algorithm 3.

Based on GLTF, we also present two variants, i.e., $GLTF_0$ and $GLTF_f$. $GLTF_0$ stands for the variant GLTF without refinement for user clustering. In particular, both the global model and local models are learned jointly per user weight g_m . However, the initial assignment of users to subsets via an external clustering method remains fixed. In our experiments, classic K was used to denote the clustering algorithm. $GLTF_f$ stands for the variant GLTF with fixed personalized user weight g_m . In other words, all the main steps of the $GLTF_f$ algorithm are the same as GLTF, except that no updating is applied to the parameter g_m . In our setting, we initialized the value of g_m as 0.5 in $GLTF_f$.

Comparison Results

Table 3 lists the rating prediction accuracy of the evaluated methods, where the lowest mean absolute error (MAE) of each dataset is highlighted in boldface.

The proposed methods, notably GLTF, clearly outperformed the well-established baselines in terms of MAE. The results indicated that the proposed tensor factorization-based methods jointly modeled multi-criteria rating information and can take the correlation among user, item, and criterion dimensions into account to improve prediction performance. By contrast, the aggregation function-based method (AFBM) applied support

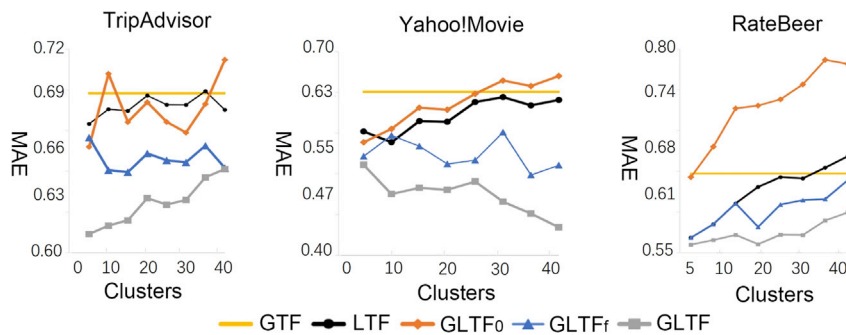


Figure 6. MAE versus Number of Clusters on Three Different Datasets

vector regression to aggregate the criteria information, which only considered the correlation between any two of the three dimensions, such as that between user and criterion, or between item and criterion. Surprisingly, tensor factorization was a good fit for MCRS, as it was an excellent way of modeling the intrinsic interactions among the three dimensions, i.e., users, items, and criteria.

We further do the experiment to evaluate the statistical significance of the differences reported in the experimental results. The p value is 0.95, which shows that our results are statistically significant.

The performance of the criteria chain method (CC) was not very stable compared with the proposed methods. The CC method relies on the tensor technique and criteria chains to exploit the correlation and dependencies among users, items, and item criteria. However, it is often difficult to accurately define the sequence of criteria in the chains because the correlation between each pair of criteria is typically complicated. In addition, it is very likely for CC to result in an accumulation of errors due to the rating prediction for current criterion depending on the prediction for previous criterion on the chain. In other words, the prediction for current criterion could be wrong if the previous predictions are incorrect.

Compared with either GTF or LTF, GLTF achieves the best performance with the lowest MAE values on all three datasets, i.e., about 0.62 on TripAdvisor, 0.51 on Yahoo!Movie, and 0.57 on RateBeer. The result demonstrates the importance of combining global model with local models. In other words, when local models and global model are combined in a user-specific way, as in the case of GLTF, we get the best performance for rating prediction. The MAE of GLTF are much lower than that of GTF (TripAdvisor about 0.07, Yahoo!Movie about 0.12, RateBeer about 0.07) and LTF (TripAdvisor about 0.06, Yahoo!Movie about 0.09, RateBeer about 0.05). The comparison between LTF and GLTF shows the benefit of adding a global model, while the comparison between GTF and GLTF shows the benefit of considering local predictive models. $GLTF_r$ and $GLTF_0$ are two variants of GLTF. The improvement of GLTF over $GLTF_r$ displays the effect of adding user-specific weight g_m , while the improvement of GLTF over $GLTF_0$ demonstrates the benefit of allowing users to switch subsets.

As described in Table 4, the MAE of $GLTF_0$ is a little higher than that of LTF. This is because the assignment of users to subsets remains fixed once user subsets were initialized in $GLTF_0$. If the initialization for user subsets happens to be inappropriate, we

learning local predictive models for individual user subsets can capture the differences of users' preferences effectively and improve the rating prediction.

To further evaluate the Top-K item recommendation, the experimental results in term of NDCG@K obtained from three datasets are shown in Figure 5, where K varies from 2 to 10. A similar conclusion can be drawn from Figure 5 demonstrating that GLTF achieves the best performance of three datasets for all cases.

Effect of Clustering Methods

We can use either existing clustering algorithms or a random partition method to initialize user subsets in GLTF. In other words, the performance of GLTF is not dependent on the clustering algorithm. To verify this, we compare the performance of GLTF under two different settings, i.e., using K-means clustering to initialize user subsets or simply splitting users into subsets at random. Figure 4 shows the performance of GLTF with different user-subset initialization methods, respectively. In Figure 4 we can see that, when the number of iterations increases, GLTF achieves comparable performance under the two different initialization settings on the TripAdvisor, Yahoo!Movie, and RateBeer datasets.

The experimental results from three datasets are similar. The gap between the curves of the two initialization clustering methods is large when the number of iterations is relatively small. This is expected, as in the first few iterations the assignment of users to optimal subsets has not yet been performed. Thus, the local models learned based on the user subsets by the clustering method are more meaningful than that with random initialization. However, as the iterations progress, much better allocation of users to subsets can be achieved by GLTF. We see that the MAE of GLTF with random initialization drops quickly and then reaches the converged state. As shown in the figures, the performance of GLTF for rating prediction is very similar for the two different initialization methods for user subsets on the datasets. This is because during training, GLTF is able to iteratively assign users to various subsets and then generate optimal clustering results upon completion. We conclude that our GLTF method was able to learn robust local models in addition to a global predictive model, even with random initialization for user subsets. It is worth noting that, when starting from random assignment of user subsets, the proposed GLTF may need more iterations in order to achieve a satisfactory performance.

may learn undesired local predictive models. As a result, the performance of $GLTF_0$ for rating prediction drops, as the ratings are predicted by using both global and local models.

Comparing LTF with GTF, we find that the performance of LTF is much better than that of GTF. This suggests that

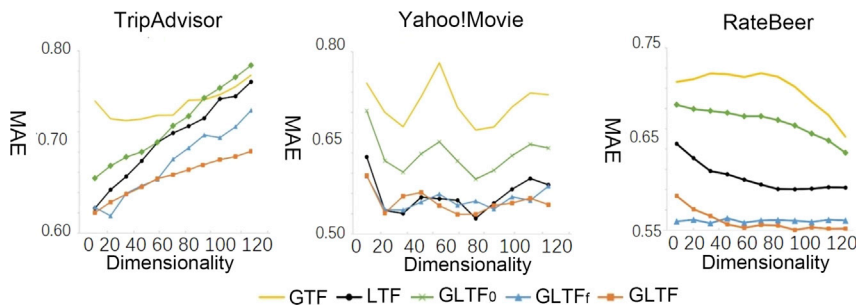


Figure 7. MAE versus Dimensionality of Latent Factor Space on Three Different Datasets

Effect of Number of Clusters

Figure 6 shows how the number of user clusters affects the performance of the proposed methods. We can see that GLTF outperforms all the other methods for all the numbers of user clusters. On the TripAdvisor and RateBeer datasets, GLTF achieves its best performance when the number of clusters is about 5. For Yahoo!Movie dataset, the best performance is achieved when the number of clusters is about 40. This is because the densities of the TripAdvisor dataset (0.21%) and RateBeer dataset (0.27%) are much lower than that of the Yahoo!Movie dataset (1.87%). When the density of dataset is low, if the number of user subsets is large, the neighbors of the targeted user in the same subset would be scarce. As a result, the prediction accuracy of local models would be reduced.

Effect of Dimensionality of Latent Factor Space

Figure 7 shows how the dimensionality of latent factor space affects the performance of the proposed methods. GLTF outperforms the other methods across almost all given dimensionality values. Specifically, all the proposed methods tend to achieve the best MAE when mapping users, items, and criteria to the latent factor space of smaller dimensionality (e.g., 10) on the TripAdvisor dataset. On the Yahoo!Movie dataset, mapping users, items, and criteria to the latent space of medium dimensionality (e.g., 70) is helpful for the methods to attain decent performance. In contrast, with increasing the dimensionality, almost all the methods improve the performance for rating prediction.

Interplay between Global and Local Predictive Models

To discover how the local models and global model affect the rating prediction, we analyzed personalized weights g_m , which control the interplay between the global and local predictive models. As shown in Equation 10, g_m varies from 0 to 1.

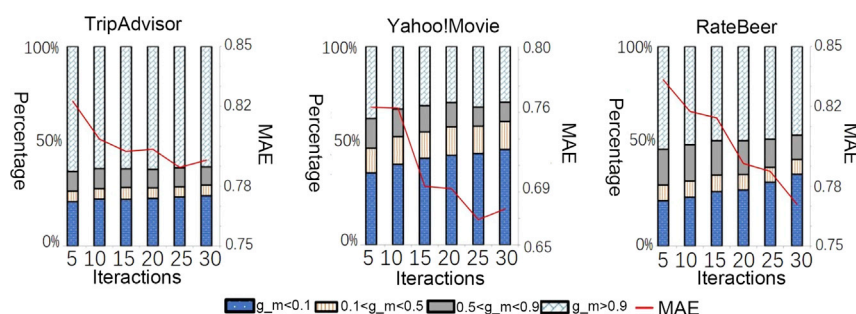


Figure 8. The Interplay between the Global and the Local Part of the Model on Three Different Datasets

When g_m is equal to 1, only the global model is used. If g_m is equal to 0, it means that the prediction is only affected by the local models. When g_m is greater than 0.5, the global model plays a more important role than the local models in GLTF

for rating prediction and vice versa. In Figure 8, the bars indicate changes of the g_m value during the iteration and the line represents the change of the MAE value on each dataset. It can be found that, by increasing the number of iterations, the percentage of g_m which is greater than 0.5 becomes smaller, and the MAE value shows a decreasing trend. This observation suggests that the effect of local information on the models becomes greater as the iterations progress. As a result, local models have more influence on rating prediction than the global model.

DISCUSSION

In this paper, we addressed the multi-criteria recommendation problem that typically involves multiple criterion-specific ratings in addition to user-item rating data. We proposed the tensor factorization techniques, notably, GLTF, to address the problem. In GLTF, we not only learned a global predictive model from the whole user-item-criteria tensor data but also simultaneously learned multiple local models from partitioned user-subset specific subtensors of rating data. Both global and local models were then jointly employed to predict the ratings of a given user on unknown items and the criteria of the items. Experimental results with real-world data have shown that the proposed GLTF method is superior to well-established baseline methods for tackling the multi-criteria recommendation problem.

More specifically, this study provides four important contributions: (1) A principal tensor factorization method was developed to leverage additional criterion-specific ratings in addition to existing user-item rating data for better existing user-item rating data for better recommendation; (2) a new unified global and local tensor factorization framework is proposed, which can jointly learn a global predictive model and multiple local predictive models for the purposes of recommendation; (3) our proposed GLTF

method is adept at discovering the overall structure of the whole rating tensor while also capturing diverse rating behaviors of users in individual subtensors; and (4) extensive experiments have been conducted with real-life data to validate the value of GLTF as a way to resolve certain well-known issues associated with multi-criteria recommendation.

A significant amount of work needs to be conducted in the future. We plan to obtain more data for a larger dataset for evaluation. At the same time, we realize that the sparsity problem is a very important issue, and we will further deliberate on improving the model to mitigate this problem.

EXPERIMENTAL PROCEDURES

This section presents the experimental procedures used to evaluate the proposed model for multi-criteria recommendation with real-world data.

Resource Availability

Lead Contact

H.Y. takes responsibility for the Lead Contact role. Her email address is yhn6@bit.edu.cn.

Materials Availability

The study did not generate new unique reagents.

Data and Code Availability

To evaluate the algorithms, we used three different datasets from TripAdvisor,⁵⁰ Yahoo!Movie,²⁵ and RateBeer,⁵¹ as shown in Table 3.

TripAdvisor is the largest travel site in the world, where users can use the 1-to-5 star rating system to rate four criteria of hotels, including value, location, service, and overall (i.e., special criteria). After cleaning, there were 23,066 records given by 6,134 users based on four criteria for 1,763 hotels. Each user gave at least two ratings. The sparsity level of the dataset was around 99.79%. The Yahoo!Movies dataset, except for movie ID, user ID, and rating, provides the gender and ages of the users. After cleaning, there were 50,673 records given by 1,827 users based on five criteria for 1,479 movies, where the five criteria are story, acting, direction, visual effects, and overall. The ratings vary from 1 to 13. Each user rated at least ten movies. The sparsity level of the dataset was around 98.13%. The RateBeer dataset includes users' evaluation of beers. After cleaning, there were 48,605 records given by 3,630 users on five criteria, namely appearance, aroma, palate, taste, and overall, for 4,896 beers. Each user rated at least five beers. The sparsity level of the dataset was around 99.73%.

Experimental Setting

Performance Metric

In experiments, 5-fold cross-validation was applied to each dataset, and MAE and normalized discounted cumulative gain (NDCG) were adopted to evaluate the recommendation performance:

$$MAE = \frac{\sum_{i=1}^N |p_{.i} - q_{.i}|}{N}, \quad (\text{Equation 14})$$

where N is the number of pairs of observed ratings $p_{.i}$ and predicted ratings $q_{.i}$ in the test set. Note that the lower the MAE, the better the recommendation performance.

$$NDCG@K = Z_K \sum_{i=1}^K \frac{2^r - 1}{\log_2(i+1)}, \quad (\text{Equation 15})$$

where Z_K ensures a value of 1 for the perfect ranking result and r_i is the graded significance of the item at position i .

Baseline Methods

We validated our proposed methods, notably $GLTF$, as well as the variants, $GLTF_0$ and $GLTF_r$, to produce multi-criteria recommendations. Note that GTF can be treated as a representative of normal tensor factorization baseline. We also compared the proposed methods with three other well-established baseline methods, including AFBM¹⁵ and CC.²² In particular, the AFBM em-

ploy a matrix factorization to factor the observed user-criterion rating data, then uses the learned model to estimate the ratings of a user on individual criteria (excluding special overall criterion). Next, it applies a support vector regression to aggregate the estimated criterion ratings for predicting overall ratings. CC attempts to leverage the dependency among multiple criteria for rating prediction.

Parameter Settings

For the purposes of this study, the value of regularization parameters β_G , β_L , α_G , and α_L were set as 0.1, and initialized $g_{.m} = 0.5$ in Algorithm 3 and method $GLTF_0$. For the TripAdvisor dataset, we set the learning rate $\lambda = 0.01$, the dimensionality of latent factor space $D = 70$, the number of iterations as 50, and the number of subsets $K = 5$ in Algorithms 2 and 3 and method $GLTF_0$, $GLTF_r$. For Yahoo!Movie dataset, we set the learning rate $\lambda = 0.005$, the dimensionality of latent factor space $D = 80$, the number of iterations as 30, and the number of subsets K was set as 5 in Algorithm 2, method $GLTF_0$, $GLTF_r$, and 40 in Algorithm 3. For RateBeer dataset, we set the learning rate $\lambda = 0.001$, the dimensionality of latent factor space $D = 80$, the number of iterations as 80, and the number of subsets K was set as 5 in Algorithms 2 and 3 and methods $GLTF_0$, $GLTF_r$.

ACKNOWLEDGMENTS

This research was supported by grants from the National Key Research and Development Program of China (2016YFC0803000), and Science and Technology Innovation Research Project of The Ministry of Science and Technology of China (ZLY201970, ZLY201976-02).

AUTHOR CONTRIBUTIONS

S.W. and H.Y. conceived, designed, and coordinated the project. H.Y., J.Y., and Z.C. performed all experimental work. S.W., H.Y., J.Y., J.G., and Z.C. wrote the manuscript. S.W., H.Y., J.Y., Z.C., J.G., and Z.H. revised the manuscript and were involved in the discussion of the work.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: December 18, 2019

Revised: March 19, 2020

Accepted: March 30, 2020

Published: May 8, 2020

REFERENCES

1. Palmisano, C., Tuzhilin, A., and Gorgoglione, M. (2008). Using context to improve predictive modeling of customers in personalization applications. *IEEE Trans. Knowl. Data Eng.* 20, 1535–1549.
2. Rendle, S., Gantner, Z., Freudenthaler, C., and Schmidt-Thieme L. (2011). Fast context-aware recommendations with factorization machines. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 635–644.
3. Bhargava, P., Phan, T., Zhou, J., and Lee, J. (2015). Who, what, when, and where: multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data. In Proceedings of the 24th International Conference, pp.130–140. <https://doi.org/10.1145/2736277.2741077>.
4. Yuan, Q., Cong, G., Zhao, K., Ma, Z., and Sun, A. (2015). Who, where, when, and what: a nonparametric Bayesian approach to context-aware recommendation and search for twitter users. *ACM T. Inf. Sys.* 33, <https://doi.org/10.1145/2699667>.
5. Wu, S., Liu, Q., Wang, L., and Tan, T. (2016). Contextual operation for recommender systems. *IEEE T. Knowl. Data Eng.* 28, 2000–2012.
6. Ishanka, U.A.P., and Yukawa, T. (2018). An analysis of emotion and user behavior for context-aware recommendation systems using pre-filtering and tensor factorization techniques. *Glob. J. Comp. Sci. Tech. D Neu. Art. Intel.* 18, 4–16.

7. Zheng, Y. (2018). Context-aware mobile recommendations by a novel post-filtering approach. In Proceedings of the 31st AAAI International Florida Artificial Intelligence Research Society Conference.
8. Zheng, Y., and Jose, A.A. (2019). Context-aware recommendations via sequential predictions. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pp.2525–2528 <https://doi.org/10.1145/3297280.3297639>.
9. Zheng, Y., Shekhar, S., Anna Jose, A., and Rai, S.K. (2019). Integrating context-awareness and multi-criteria decision making in educational learning. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pp. 2453–2460.
10. McAuley, J., and Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating dimensions with review text. In Proceedings of the 7th ACM Conf Recommender Systems, pp.165–172.
11. Bao, Y., Fang, H., and Zhang, J. (2014). TopicMF simultaneously exploiting ratings and reviews for recommendation. In Proceedings of the 28th AAAI Conference on Artificial Intelligence, pp.2–8.
12. Zhang, W., Yuan, Q., Han, J., and Wang, J. (2016). Collaborative multi-level embedding learning from reviews for rating prediction. In Proceedings of the 25th International Joint Conference on Artificial Intelligence, pp. 2986–2992.
13. Zheng, L., Noroozi, V., and Yu, P.S. (2017). Joint deep modeling of users and items using reviews for recommendation. In Proceedings of the 10th ACM International Conference on Web Search and Data Mining, pp. 425–434.
14. Cao, D., He, X., Nie, L., Wei, X., Hu, X., Wu, S., and Chua, T.S. (2017). Cross-platform app recommendation by jointly modeling ratings and texts. *ACM Trans. Inf. Sys.* 35, 37.
15. Adomavicius, G., and Kwon, Y.O. (2007). New recommendation techniques for multicriteria rating systems. *IEEE Intel. Sys.* 22, 48–55.
16. Lakiotaki, K., Matsatsinis, N.F., and Tsoukias, A. (2011). Multi-criteria user modeling in recommender systems. *IEEE Intel. Sys.* 26, 64–76.
17. Liu, L., Mehandjiev, N., and Xu, D.L. (2011). Multi-criteria service recommendation based on user criteria preferences. In Proceedings of the 5th ACM Conference of Recommender Systems, pp. 77–84.
18. Mikeli, A., Apostolou, D., and Despotis, D. (2013). A multi-criteria recommendation method for interval scaled ratings. In Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence, pp.9–12.
19. Syamala, R.S., Patra, B.K., and Hernando, A. (2017). Multi-criteria recommendations through preference learning. In Proceedings of the 4th ACM IKDD Conf. Data Sci., pp.1–11. DOI: 10.1145/3041823.3041824.
20. Lakiotaki, K., Tsafarakis, S., and Matsatsinis, N. (2008). Uta-rec: a recommender system based on multiple criteria analysis. In Proceedings of the 2008 ACM Conference on Recommender Systems, pp. 219–226.
21. Jannach, D., Karakaya, Z., and Gedikli, F. (2012). Accuracy improvements for multi-criteria recommender systems. In Proceedings of the 13th ACM Conference on Electronic Commerce, pp. 674–689.
22. Zheng, Y. (2017). Criteria chains: a novel multi-criteria recommendation approach. In Proceedings of the 22nd International Conference on Intelligent User Interfaces, pp. 29–33.
23. Hamada, M., Abdulsalam, L., and Mohammed, H. (2018). Adaptive genetic algorithm for improving prediction accuracy of a multi-criteria recommender system. In Proceedings of the IEEE 12th International Symposium on Embedded Multicore/Many-core Systems-on-Chip, pp. 79–86.
24. Zheng Y. (2019). Utility-based multi-criteria recommender systems. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing. pp. 2529–2531.
25. Sahoo, N., Krishnan, R., Duncan, G., and Callan, J. (2012). Research note—the Halo effect in multicomponent ratings and its implications for recommender systems: the case of yahoo!Movies. *Inf. Sys. Res.* 23, 231–246.
26. Nilashi, M., Ibrahim, O.B., and Ithnin, N. (2014). Hybrid recommendation approaches for multi-criteria collaborative filtering. *Exp. Sys. Appl.* 41, 3879–3900.
27. Hamad, M., Nkiruka, O., and Hassan, M. (2018). A fuzzy-based approach for modelling preferences of users in multi-criteria recommender systems. In Proceedings of the IEEE 12th International Symposium on Embedded Multicore/Many-core Systems-on-Chip, pp. 87–94.
28. Li, Q., Wang, C., and Geng, G. (2008). Improving personalized services in mobile commerce by a novel multicriteria rating approach. In Proceedings of the 17th International Conference on World Wide Web, pp. 1235–1236.
29. Hassan, M., and Hamada, M. (2017). A neural networks approach for improving the accuracy of multi-criteria. *Recom. Sys. App. Sci.* 7, 868.
30. Tallapally, D., Sreepada, R.S., and Patra, B.K. (2018). User preference learning in multi-criteria recommendations using stacked auto encoders. In Proceedings of the 12th ACM Conference on Recommender Systems, pp. 475–479.
31. Ma, H., Yang, H., Lyu, M.R., and King, I. (2008). Sorec: social recommendation using probabilistic matrix factorization. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 931–940.
32. Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer* 42, 30–37.
33. Jamali, M. and Ester, M. (2010). A matrix factorization technique with trust propagation for recommendation in social networks. In Proceedings of the 4th ACM Conference on Recommender Systems, pp. 135–142.
34. Beutel, A., Chi, E.H., Cheng, Z., Pham, H., and Anderson, J. (2017). Beyond globally optimal: focused learning for improved recommendations. In Proceedings of the 26th International Conference on World Wide Web, pp. 203–212.
35. Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 426–434.
36. Papalexakis, E.E., Faloutsos, C., and Sidiropoulos, N.D. (2017). Tensors for data mining and data fusion: models, applications, and scalable algorithms. *ACM Trans. Intel. Sys. Tech.* 8, 1–44.
37. Rendle, S., Marinho, L.B., Nanopoulos, A., and Schmidt-Thieme, L. (2009). Learning optimal ranking with tensor factorization for tag recommendation. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 727–736.
38. Karatzoglou, A., Amatriain, X., Baltrunas, L., and Oliver, N. (2010). Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In Proceedings of the 4th ACM Conference on Recommender Systems, pp.79–86.
39. Zheng, V.W., Cao, B., Zheng, Y., Xie, X., and Yang, Q. (2010). Collaborative filtering meets mobile recommendation: a user-centered approach. In Proceedings of the 24th AAAI Conference on Artificial Intelligence, pp.236–241.
40. Zheng, V.W., Zheng, Y., Xie, X., and Yang, Q. (2012). Towards mobile intelligence: learning from GPS history data for collaborative recommendation. *Art. Intel.* 184–185, 17–37.
41. Yao, L., Sheng, Q.Z., Qin, Y., Wang, Shemshadi A. and He, Q. (2015). Context-aware point-of-interest recommendation using tensor factorization with social regularization. In Proceedings of the 38th International ACM SIGIR Conference on Research Development in Information Retrieval, pp. 1007–1010.
42. Christakopoulou, E., and Karypis, G. (2016). Local item-item models for Top-N recommendation. In Proceedings of the 10th ACM Conference on Recommender Systems, pp. 67–74.
43. Xu, B., Bu, J. Chen, C., and Cai, D. (2012). An exploration of improving collaborative recommender systems via user-item subgroups. In Proceedings of the 21st International Conference on World Wide Web, pp. 21–30.

44. Lee, J., Kim, S., Lebanon, G., and Singer, Y. (2013). Local low-rank matrix approximation. In *Proc. Int. Conf. Mach. Lear.*, pp. 82–90.
45. Lee, J., Bengio, S., Kim, S., Lebanon, G., and Singer, Y. (2014). Local collaborative ranking. In *Proceedings of the 23rd International Conference on World Wide Web*, pp. 85–96.
46. Beutel, A., Murray, K., Faloutsos, C., and Smola, A.J. (2014). Cobafi: collaborative Bayesian filtering. In *Proceedings of the 23rd International Conference World Wide Web*, pp. 97–108.
47. Wu, Y., Liu, X., Xie, M., Ester, M., and Yang, Q. (2016). CCCF: improving collaborative filtering via scalable user-item co-clustering. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*, pp. 73–82.
48. Wang, K., Zhao, W.X., Peng, H., and Wang, X. (2016). Bayesian probabilistic multi-topic matrix factorization for rating prediction. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pp. 3910–3916.
49. Bottou, L., and Bousquet, O. (2007). The tradeoffs of large scale learning. In *Proc. Int. Conf. Neu. Inf. Proces. Syst.*, pp. 161–168.
50. Wang, H., Lu, Y., and Zhai, C. (2011). Latent aspect rating analysis without aspect keyword supervision. In *Proc. 17th ACM SIGKDD Conf. Knowl. Disc. Data Min.*, pp. 618–626.
51. McAuley, J., and Leskovec, J. (2013). From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proc. Int. Conf. WWW*, pp. 897–908.