Improving Rating Prediction in Multi-criteria Recommender Systems via a Collective Factor Model

Ge Fan, Chaoyun Zhang, Junyang Chen, *Member, IEEE,*
Paul Li, Yingjie Li, Victor C. M. Leung, *Life Fellow, IEEE*

***Abstract*—Existing recommendation methods usually train several independent modules for each rating information instead of an end-to-end manner. Therefore, these methods may be incapable of collaborative learning leading to sub-optimal results in predicting users' overall interests. The main disadvantage of those two-stage methods is that the overall rating heavily relies on the predicted sub-ratings, and the predictive error of sub-ratings is accumulated during the regression step. Moreover, the regression model is trained with unbiased sub-ratings but used with biased predictive sub-ratings. Meanwhile, the separate training pattern induces more training overhead. To address these problems, we propose a collective model to predict a user's overall rating, which can learn each of the multi-criteria sub-scores simultaneously in an end-to-end manner. This enables the proposed method to improve its prediction quality by transferring the knowledge of criterion to the domain of overall ratings. It reduces the dependence of the predicted score of a specific criterion, making the overall system more robust. In addition, our end-to-end method avoids learning the regress part directly from the unbiased sub-ratings, improving the performance of the overall model. Experiments on three real-world datasets show that our proposed architecture achieves up to 13.14% lower prediction error over baseline approaches.**

***Index Terms*—Latent factor model, Multi-criteria recommender system, Collaborative filtering, Collective matrix factorization**

I. Introduction

In a recommender system, users can give multi-criteria ratings for each item, e.g., an overall and detailed ratings of this item's attributes. Thus, it is important to find out an effective method for exploiting users' such rating information to predict their interests. Fortunately, recent research on the multi-criteria recommender system allows to estimate the overall scores of an item by weighting the sub-score of each of its attribute, to reflect a user's preference of the item. As it weights an item using multi-dimensional perspectives, the multi-criteria recommender systems perform better over single-criterion counterparts in terms of the predictive accuracy, and are used more frequently in both industry and academia, e.g.*BeerAdvocate* [1]–[3] *TripAdvisor Yahoo!Movies* [4], [5].

One crucial challenge for the multi-criteria recommender system is integrating the user's multi-criteria ratings to predict the overall rating of the item. To this end, traditional multi-criteria recommender systems usually predict overall rating of an item to a user in two separate stages (e.g. [1], [6]). They first predict ratings of each attribute of the item, and then employ a separate regression model to integrate those scores to obtain an overall rating. However, there are still several problems, making separate stage methods challenging:

1) Low performance of sub-scored predictive model. The overall rating heavily relies on the predicted sub-scored of each attribute, which is particular difficult to predict. A possible reason is that each sub-scored predictive model are trained separately, the knowledge of each criterion cannot be transferred and shared. As a result, inaccurate prediction of the sub-scores will significantly affect the overall system, lowering the performance of the entire recommender systems.
2) Bias in regression model. The regression model are trained on unbiased rated sub-rating, while are utilized to predict the overall rating with predicted biased sub-rating. Such a gap may hurt the generalization performance of regression model.

To overcome these issues, we propose an end-to-end collective filtering method in the multi-criteria rating system, to integrate the sub-scores and overall rating into a unique architecture. Specifically, we employ a latent factor model [7] to jointly learn users' overall preferences, as well as their preferences on each criterion of an item. The sub-scores of each criterion are treated as latent variables of the model, which are hidden and do not need to be estimated in a separate process. The knowledge of multi-criteria ratings can be shared and transferred to the domain of overall ratings, which reduces the dependency of the predicting of sub-ratings and adaptively captures the latent information from different criteria. In addition, the regression part of our method is trained with the predicted biased sub-ratings, which improves the generalization performance as it is the same as the pre-

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62102265, by the Open Research Fund from Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ) under Grant No. GML-KF-22-29, by the Natural Science Foundation of Guangdong Province of China under Grant No. 2022A1515011474, and by Shenzhen Talents Special Project - Guangdong Provincial Innovation and Entrepreneurship Team Supporting Project under Grant No. 2021344612. (Corresponding author: Junyang Chen)

Ge Fan and Yingjie Li are with the Lightspeed & Quantum Studios, Tencent Inc., Shenzhen, China, {gefan,wallaceyjli}@tencent.com;

Chaoyun Zhang is with the Microsoft, Beijing, China, chaoyun.zhang@microsoft.com;

Junyang Chen and Victor C. M. Leung are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China, junyangchen@szu.edu.cn, vleung@ieee.org;

Paul Li is with the Baidu, Sunnyvale, USA, baopuli@baidu.com;

dicting step. Our proposed method makes the overall system more robust and accurate, as it does not heavily rely on the predicted score of a specific criterion.

Overall, this paper makes the following three contributions, namely:

1) We propose an end-to-end collective factor model to unify the learning processes of sub-scores and overall rating. This reduces the dependency of the score of each attribute, improving the robustness of the entire system;
2) We theoretically analyze the knowledge transferring of proposed model, and investigate the impact of different latent vector sharing schemes between users and items. Experiments show that latent information can be automatically captured in our model, since keeping both the user latent vector and the item latent vector independent outperforms over other two sharing methods;
3) We compare our proposed architectures with several baseline methods on three real-world datasets. The results show that our solution outperforms baseline approaches by achieving up to 13.14% lower prediction error over state-of-the-art approaches.

II. Related Work

To solve the task of the multi-criteria recommender system, we propose a collective factor model which is based on the collective matrix factorization [8]. Therefore, both *multi-criteria* based and *collective matrix factorization* based approaches are related to our work. In this section, we briefly review related literatures.

A. Multi-criteria recommendation approaches

A primary method to exploit the information of multi-criteria ratings is extending the user similarity calculation from single-criterion to multi-criteria. Nilashi et al. [9] exploited a fuzzy method for the calculation of similarity between users by employing the multi-criteria ratings. Kermany et al. [10] integrated the fuzzy cosine and Jaccard similarity to obtain the final similarity between users/movies.

Recently, two-stage based methods emerge and become more popular in the multi-criteria recommender system. At the first stage, the multi-criteria ratings of the target item are estimated. The overall ratings are obtained by learning the weights of each sub-score by a separate model, such as linear regression [1], support vector regression [3], [11] and neural networks [6], [12]. When the multi-criteria ratings are not explicitly expressed by users, the text reviews can be used to uncover the aspect (similar to criterion). Individual attitude on each aspect can be classified into positive, neutral or negative [12]–[14]. Such method also requires to estimate the individual attitude towards each aspect at the first stage and then predicts the overall rating.

The rating data of a multi-criteria ratings in recommender system can be represented as a 3-D tensor, where the first two dimensions represent users and items, and the third is the ratings of different criteria. Therefore, the tensor factorization and higher order singular value decomposition can be used to learn preferences of users and attributes of items [15]–[21]. For these two types of methods, the predictive model can be trained *end-to-end* and a user's overall rating is directly estimated by the model, which does not require to compute the multi-criteria ratings initially.

Deep learning-based methods have become a popular research topic, as evidenced by recent studies [22]–[29]. [23] introduces a novel approach to tackle the challenge of multi-criteria recommendation. Their method leverages the power of sparse autoencoders to balance and integrate multiple criteria, leading to improved recommendation accuracy. Research in [24] proposes a deep neural network-based matrix factorization method for filtering information in multi-criteria recommendation systems. Their approach aims to enhance recommendation accuracy and efficiency. Paper [22] presents a deep learning-based approach to multi-criteria recommendation for hotel recommendations, incorporating Dempster-Shafer theory to handle uncertainty and improve recommendation accuracy. In a different approach, [25] introduces a context-aware recommendation system. Their method incorporates contextual information to enhance recommendation accuracy and relevance. [27] investigates the effect of incorporating contextual information on the overall rating and recommendation accuracy in multi-criteria recommendation systems. Finally, work in [28] accounts for multiple stakeholder preferences and utilize multi-criteria ratings to improve recommendation performance. Overall, these studies highlight the potential of deep learning-based methods to enhance the accuracy, efficiency, and relevance of multi-criteria recommendation systems.

B. Collective matrix factorization based approaches

The collective matrix factorization (CMF) can be employed to incorporate the rating data and the auxiliary data [8], [30], when the auxiliary data is available. The collective factorization method simultaneously co-factorizes a variety of matrices when an entity participates in multiple relations. It has demonstrated superiority when integrating diverse auxiliary resources, such as social networks [31]–[34], geographical information [35]–[37] and contents of items [8], [38], as it can effectively embed those rich resources [38]–[40]. Singh et al. [8] proposed a collective matrix factorization model to improve the predictive accuracy by integrating multiple matrices. Based on the collective matrix factorization, Liu et al. [41] incorporated both explicit and implicit feedback of users to improve recommendation quality. Similarly, Yuan et al. [31], [42] exploited collective matrix factorization method to jointly model data from different sources, which improve the performance of the model. Ma et al. [32] extended the probabilistic matrix factorization approach to fuse both the user-item rating matrix and the social network. Yang et al. [43] combined the factor-based model and random walk to alleviate the sparisty problem in the recommender system. Zhao et al. [44] built user profiles by directly combining these diverse behavioral signals. The data of each behavior is represented by a matrix. A collective factorization based method is utilized to co-factorize these behavioral tensor for the recommendation. In [45], the social matrix factorization method is combined with the topic matrix factorization approach to jointly model different ratings, item reviews and social relations.

With the rapid growth and prevalence of location tracking services, increasing amount of mobile data become available. Similar to social relations, the user location matrix can be co-factorized with the user activity matrix to provide location as well as activity recommendations for users [35], [36], [46]. Zheng et al. [46] exploited the collective matrix factorization method to solve the problem of location-related queries. Zhao et al. [36] combined users' ratings with their geographical information to solve the cold-start and sparsity problem in recommender systems. When the side information of users and items is available, the collective matrix factorization based approaches can be used to embed those rich resources [38]–[40], [47]. Fang et al. [39] co-factorized the user information matrix and the item information matrix to solve the task of recommendations in online scientific communities. Saveski et al. [38] exploited items' properties and user preferences via collective embedding to solve the item cold-start problem. Liu et al. [40] proposed a non-negative matrix factorization based method to jointly model both consistent and complementary information among multi-view data.

C. Limitations of traditional methods

Although the user similarity calculated by multi-criteria ratings is in general more accurate, such methods usually suffer from the poor efficiency and low robustness, as most of existing multi-criteria based models are not trained in an end-to-end manner. As it heavily relies on the the sub-scores on different attributes, the overall accuracy of the model is highly sensitive to the its sub-processes.

To resolve this problem, we propose a collective factor model, which combines contributions of overall ratings and multi-criteria ratings in a linear manner. The experimental results demonstrate that our method is superior over existing multi-criteria based approaches in terms of accuracy of overall rating predictions.

III. The Proposed Method

In this section, we first introduce the notations used in this work in Table I. Without loss of generality, individual overall ratings can be represented as a weighted adjacent matrix $\mathbf{R}^{(0)} = \{r_{ui}^{(0)}\}_{n\times m}$, where $r_{ui}^{(0)}$ is the overall rating that user u gives to the item i. n and m are the number of users and items in the system, respectively. Similarly, an adjacent matrix $\mathbf{R}^{(\alpha)} = \{r_{ui}^{(\alpha)}\}$ can be used to represent users' ratings on criterion α. The task of a multi-criteria recommender system is estimating a user's overall rating by utilizing the ratings on each criterion.

A. Latent factor model

In this paper, the latent factor model is adopted to learn users' preferences, jointly mapping users and items to the same space with dimensionality k. We denote two latent vectors as $\mathbf{x}_u$ and $\mathbf{y}_i$, as the preference of user u and the attribute of item i respectively. The predictive score can be easily obtained by the inner product of these two latent vectors:

$$\hat{r}_{ui} = \mathbf{x}_u^T \mathbf{y}_i. \tag{1}$$

TABLE I
Notations used in this paper.

Notation	Description
n	The user number in the system
m	The item number in the system
c	The number of criteria
$\mathbf{R}^{(0)}$	The overall rating matrix
$r_{ui}^{(0)}$	The overall rating that user u gives to item i
$\widehat{r}_{ui}^{(0)}$	The predictive overall rating between user u and item i
$\mathbf{R}^{(\alpha)}$	The rating matrix $w.r.t$ criterion α
$r_{ui}^{(\alpha)}$	The rating on criterion α that user u gives to item i
$\widehat{r}_{ui}^{(\alpha)}$	The predictive rating on criterion α between user u and item i
$\mathbf{x}_u^{(0)}, \mathbf{y}_i^{(0)}$	The latent factor vector of user u and item i $w.r.t$ overall ratings
$\mathbf{x}_u^{(\alpha)}, \mathbf{y}_i^{(\alpha)}$	The latent factor vector of user u and item i for ratings on criterion α
λ_0	The regularization coefficient of the latent model $w.r.t$ overall ratings
λ_α	The regularization coefficient of the latent model for ratings on criterion α
Θ_α	The hyper-parameter controlling contributions of ratings on criterion α
w_α	The weight of the predictive score on criterion α
ε	The error variable

The latent factor model has been widely studied in recent years due to its flexibility and scalability. It is adaptable to application-specific requirements, such as biases of users and items [7]. Then, Eq. (1) can be extended as:

$$\hat{r}_{ui} = \mu + b_i + b_u + \mathbf{x}_u^T \mathbf{y}_i, \tag{2}$$

where b_u and b_i are the biases of user u and item i, respectively, and μ is the global average rating.

We choose the biased matrix factorization (BMF) as our baseline approach for the following reasons. First, it outperforms traditional recommendation algorithms such as collaborative filtering, basic matrix factorization and probabilistic matrix factorization in terms of rating prediction [48], [49]. Second, it has comparable complexity with the basic matrix factorization.

Given ratings observed, parameters in BMF model can be learned by minimizing the following loss function:

$$\min_{\mathbf{x}^*,\mathbf{y}^*,b^*} \sum_{u,i} (r_{ui} - \widehat{r}_{ui})^2 + \lambda(\|\mathbf{x}_u\|^2 + \|\mathbf{y}_i\|^2 + b_u^2 + b_i^2), \tag{3}$$

where λ is the weight of the regularization term. This can reduce the overfitting in training process. $\mathbf{x}^*$, $\mathbf{y}^*$ and b^* are the general form of $\mathbf{x}_u$, $\mathbf{y}_i$, b_u and b_i, respectively. Values of $\mathbf{x}_u$, $\mathbf{y}_i$, b_u and b_i are initialized randomly from a Gaussian distribution (e.g. mean 0 and standard deviation of 0.01), and optimized by stochastic gradient descent (SGD) approach:

$$\begin{aligned}
b_u &\leftarrow b_u + \gamma \cdot ((r_{ui} - \widehat{r}_{ui}) - \lambda \cdot b_u)\\
b_i &\leftarrow b_i + \gamma \cdot ((r_{ui} - \widehat{r}_{ui}) - \lambda \cdot b_i)\\
\mathbf{x}_u &\leftarrow \mathbf{x}_u + \gamma \cdot ((r_{ui} - \widehat{r}_{ui}) \cdot \mathbf{y}_i - \lambda \cdot \mathbf{x}_u)\\
\mathbf{y}_i &\leftarrow \mathbf{y}_i + \gamma \cdot ((r_{ui} - \widehat{r}_{ui}) \cdot \mathbf{x}_u - \lambda \cdot \mathbf{y}_i)
\end{aligned} \tag{4}$$

where γ is the learning rate.

In this section, we introduce the overall framework of the proposed collective factor model (CFM) employed for the multi-criteria recommender system. Our method simultaneously co-factorizes the overall rating matrix and multi-criteria rating matrices, which contributes the overall performance.

B. Latent factor model for overall ratings

With the overall rating matrix $\mathbf{R}^{(0)}$, the biased matrix factorization model can be learned by minimizing the loss function:

$$\min_{\mathbf{x}^*,\mathbf{y}^*,b^*} \sum_{u,i} (r_{ui}^{(0)} - \hat{r}_{ui}^{(0)})^2 + \lambda_0(\|\mathbf{x}_u^{(0)}\|^2 + \|\mathbf{y}_i^{(0)}\|^2 + (b_u^{(0)})^2 + (b_i^{(0)})^2), \tag{5}$$

where λ_0 is the regularization parameter. $\mathbf{x}_u^{(0)}$, $\mathbf{y}_i^{(0)}$, $b_u^{(0)}$ and $b_i^{(0)}$ are model parameters with respect to overall ratings. $\mathbf{x}^*$, $\mathbf{y}^*$ and b^* are general forms of $\mathbf{x}_u^{(0)}$, $\mathbf{y}_i^{(0)}$, $b_u^{(0)}$ and $b_i^{(0)}$. $\hat{r}_{ui}^{(0)}$ is the predictive score defined as follow:

$$\hat{r}_{ui}^{(0)} = \mu^{(0)} + b_i^{(0)} + b_u^{(0)} + (\mathbf{x}_u^{(0)})^T \mathbf{y}_i^{(0)}. \tag{6}$$

We denote the parameter set of the latent factor model as $\mathbf{S}^{(0)}$ for overall ratings.

C. Latent factor model for multi-criteria ratings

By using ratings on criterion α, we can also train a biased matrix factorization model by minimizing the loss function:

$$\begin{aligned}\min_{\mathbf{x}^*,\mathbf{y}^*,b^*} \sum_{u,i} (r_{ui}^{(\alpha)} - \hat{r}_{ui}^{(\alpha)})^2 + \lambda_\alpha(\|\mathbf{x}_u^{(\alpha)}\|^2 \\ + \|\mathbf{y}_i^{(\alpha)}\|^2 + (b_u^{(\alpha)})^2 + (b_i^{(\alpha)})^2),\end{aligned} \tag{7}$$

where $\mathbf{x}_u^{(\alpha)}$, $\mathbf{y}_i^{(\alpha)}$, $b_u^{(\alpha)}$ and $b_i^{(\alpha)}$ are model parameters with respect to ratings on criterion α. $\hat{r}_{ui}^{(\alpha)}$ is the predictive score on criterion α defined as follow:

$$\hat{r}_{ui}^{(\alpha)} = \mu^{(\alpha)} + b_i^{(\alpha)} + b_u^{(\alpha)} + (\mathbf{x}_u^{(\alpha)})^T \mathbf{y}_i^{(\alpha)}. \tag{8}$$

Similarly, we use $\mathbf{S}^{(\alpha)}$ to represent the parameter set of the latent factor model with respect to the ratings on criterion α. When multi-criteria ratings are unavailable, only Eq. (6) is exploited to predict missing values in the overall ratings. In this paper, both overall ratings and multi-criteria ratings are taken into consideration to train the predictive model.

D. Collective factor model

When both overall ratings and multi-criteria ratings are available, we combine Eq. (5) and Eq. (7) in a unified framework. We assume that the final predictive score of the overall rating $\hat{r}_{ui}$ is a linear combination of the sub-rating of each criterion, with the predicted overall rating, i.e.:

$$\hat{r}_{ui} = \hat{r}_{ui}^{(0)} + \sum_{\alpha=1}^{c} w_\alpha \hat{r}_{ui}^{(\alpha)} + \varepsilon, \tag{9}$$

where w_α is the weight of the predictive score $\hat{r}_{ui}^{(\alpha)}$ on criterion α, and ε is the error variable. The term $\hat{r}_{ui}^{(0)}$ can be viewed as a modified version of the overall rating, $\hat{r}_{ui}$. However, we contend that $\hat{r}_{ui}^{(0)}$ possesses the ability to predict the final overall ratings without requiring additional criterion knowledge. Notably, the proposed model reduces to Bayesian matrix factorization (BMF) when all w_α are equal to 0, and it can capture criterion knowledge when w_α values are greater than 0. The adaptability of w_α can be learned during training, enabling our model to perform well irrespective of the relevance of domain information. The following experimental results validate this argument. The final objective function of the collective factor model includes three parts: the loss of overall ratings, the loss of multi-criteria ratings and the regularization term, defined as:

$$\begin{aligned}L &= L_0 + \sum_{c=1}^{\alpha} \Theta_\alpha L_\alpha + \Phi \\ &= \min_{\mathbf{x}^*,\mathbf{y}^*,b^*} \sum_{u,i} (r_{ui}^{(0)} - \hat{r}_{ui})^2 + \sum_{\alpha=1}^{c} \Theta_\alpha \sum_{u,i} (r_{ui}^{(\alpha)} - \hat{r}_{ui}^{(\alpha)})^2 + \Phi,\end{aligned} \tag{10}$$

where c is the number of criteria and Θ_α is a hyper-parameter that controls contributions of ratings on criterion α. Φ is the regularization term to reduce overfitting:

$$\begin{aligned}\Phi &= \lambda_0(\|\mathbf{x}_u^{(0)}\|^2 + \|\mathbf{y}_i^{(0)}\|^2 + (b_u^{(0)})^2 + (b_i^{(0)})^2) \\ &+ \sum_{\alpha=1}^{c} \lambda_\alpha(\|\mathbf{x}_u^{(\alpha)}\|^2 + \|\mathbf{y}_i^{(\alpha)}\|^2 + (b_u^{(\alpha)})^2 + (b_i^{(\alpha)})^2 + w_\alpha^2).\end{aligned} \tag{11}$$

We apply the SGD approach to train the overall model and obtain the optimal parameters in Eq. (10). For a given overall rating $r_{ui}^{(0)}$ and corresponding multi-criteria ratings $r_{ui}^{(\alpha)}$ ($\alpha = 1, 2, ..., c$), the model parameters are updated as follows:

$$\begin{aligned}
\mathbf{x}_u^{(0)} &\leftarrow \mathbf{x}_u^{(0)} + \gamma(e_{ui}\mathbf{y}_i^{(0)} - \lambda_0 \mathbf{x}_u^{(0)}) \\
\mathbf{y}_i^{(0)} &\leftarrow \mathbf{y}_i^{(0)} + \gamma(e_{ui}\mathbf{x}_u^{(0)} - \lambda_0 \mathbf{y}_i^{(0)}) \\
b_u^{(0)} &\leftarrow b_u^{(0)} + \gamma(e_{ui} - \lambda_0 b_u^{(0)}) \\
b_i^{(0)} &\leftarrow b_i^{(0)} + \gamma(e_{ui} - \lambda_0 b_i^{(0)}) \\
\mathbf{x}_u^{(\alpha)} &\leftarrow \mathbf{x}_u^{(\alpha)} + \gamma(e_{ui} w_\alpha \mathbf{y}_i^{(\alpha)} + e_{ui}^{(\alpha)} \Theta_\alpha \mathbf{y}_i^{(\alpha)} - \lambda_\alpha \mathbf{x}_u^{(\alpha)}) \\
\mathbf{y}_i^{(\alpha)} &\leftarrow \mathbf{y}_i^{(\alpha)} + \gamma(e_{ui} w_\alpha \mathbf{x}_u^{(\alpha)} + e_{ui}^{(\alpha)} \Theta_\alpha \mathbf{x}_u^{(\alpha)} - \lambda_\alpha \mathbf{y}_i^{(\alpha)}) \\
b_u^{(\alpha)} &\leftarrow b_u^{(\alpha)} + \gamma(e_{ui} w_\alpha + e_{ui}^{(\alpha)} \Theta_\alpha - \lambda_\alpha b_u^{(\alpha)}) \\
b_i^{(\alpha)} &\leftarrow b_i^{(\alpha)} + \gamma(e_{ui} w_\alpha + e_{ui}^{(\alpha)} \Theta_\alpha - \lambda_\alpha b_i^{(\alpha)}) \\
w_\alpha &\leftarrow w_\alpha + \gamma(e_{ui} \hat{r}_{ui}^{(\alpha)} - \lambda_\alpha w_\alpha) \\
\varepsilon &\leftarrow \varepsilon + \gamma e_{ui}
\end{aligned} \tag{12}$$

where $e_{ui} = r_{ui}^{(0)} - \hat{r}_{ui}, e_{ui}^{(\alpha)} = r_{ui}^{(\alpha)} - \hat{r}_{ui}^{(\alpha)}$. γ is the learning rate.

From Eq. (12), parameters are updated by both overall ratings and multi-criteria ratings. This means that the knowledge of multi-criteria ratings can be transferred to the domain of overall ratings. Knowledge sharing is widely employed in traditional recommender systems, e.g. social recommender systems [31], content-based recommender systems [38], [50] and Point-of-Interest recommender systems [35].

A simple way to share the knowledge is utilizing a common latent vector between different users or/and items. We employ

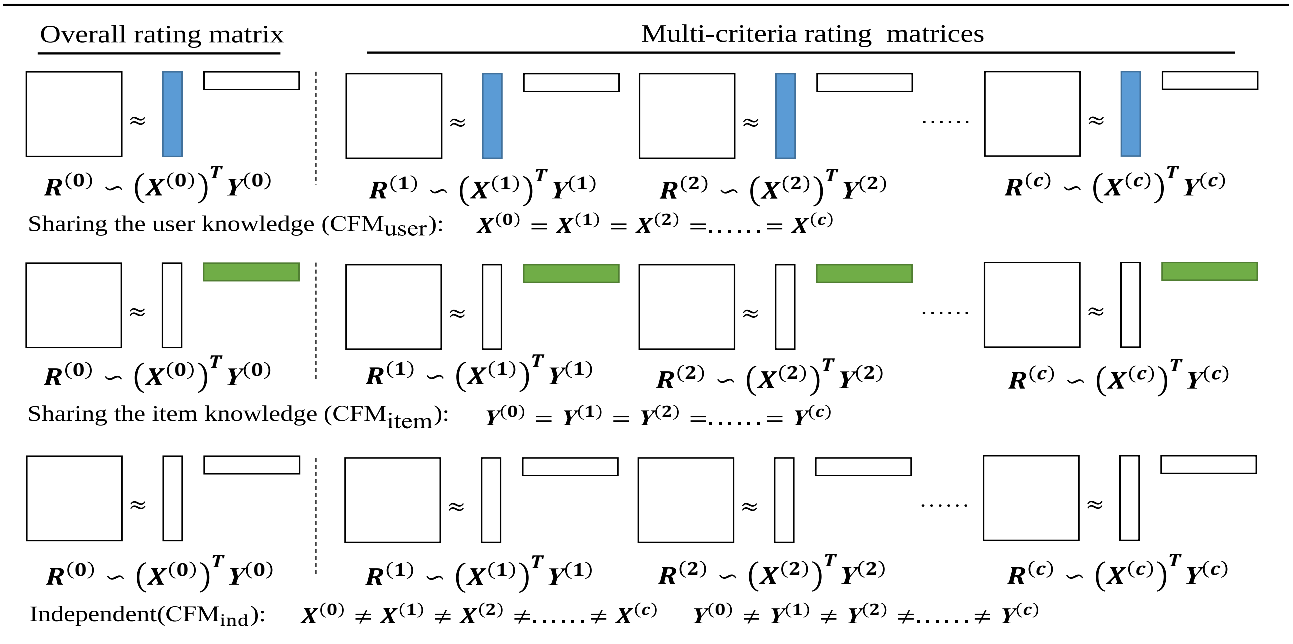

Fig. 1. The illustration of CFM method.

three variants of knowledge sharing schemes in our CFM, namely (1) sharing knowledge between overall and all criterion ratings for a user (CFM_{user}); (2) sharing knowledge between overall and all criterion rating for a item (CFM_{item}); and (3) sharing knowledge implicitly for a user or an item (CFM_{ind}). We show the structures of these three different knowledge sharing methods in Figure 1. Specifically,

1) The CFM_{user} shares the user knowledge by forcing $\mathbf{x}_u^{(0)} = \mathbf{x}_u^{(1)} = ... = \mathbf{x}_u^{(c)}$, while freeing other parameters independent.
2) The CFM_{item} shares the item knowledge by keeping $\mathbf{y}_i^{(0)} = \mathbf{y}_i^{(1)} = ... = \mathbf{y}_i^{(c)}$, while other user latent vectors are not constrained.
3) The CFM_{ind} does not apply any constraints on the user latent vector and the item latent vector, while keeping $\mathbf{x}_u^{(0)} \neq \mathbf{x}_u^{(1)} \neq ... \neq \mathbf{x}_u^{(c)}$ and $\mathbf{y}_i^{(0)} \neq \mathbf{y}_i^{(1)} \neq ... \neq \mathbf{y}_i^{(c)}$. This introduces more flexibility to the model.

IV. Experiments

In order to evaluate the performance of the proposed method, we compare it with 8 baseline approaches on 3 benchmark multi-criteria datasets.

A. Dataset

The dataset we used include *TripAdvisor*, *Yahoo!Movies* and *BeerAdvocate*. Specifically,

- **TripAdvisor.** The *TripAdvisor* dataset was released by Wang et al. [51], which consists of $1,725$ users, $3,347$ items and $29,962$ ratings. This dataset is very comprehensive in terms of criteria, i.e. *Service*, *Rooms*, *Sleep Quality*, *Location*, *Cleanliness* and *Value*. The rating scale of TripAdvisor dataset is $1-5$.
- **Yahoo!Movies.** The *Yahoo!Movies* dataset was collected by Jannach et al. [52] which embraces four criteria, including (*Acting*, *Direction*, *Story* and *Visuals*) for users. Consistent with prior studies [52], [53], we adopt a common practice of converting the ratings, which were originally provided on a 13-point scale (ranging from A+ to F), to the standard 1-5 rating scale.
- **BeerAdvocate.** The *BeerAdvocate* website allows a user to rate four attributes (*Aroma*, *Apprearance*, *Palate* and *Taste*) of beer. By removing those inactive users, the pre-processed dataset includes $3,238$ users, $2,893$ items and $88,242$ ratings. The BeerAdvocate also employs a $1-5$ scaling system for both overall and multi-criteria ratings. However, the difference between two consecutive levels is 0.5, indicating that users can evaluate beers on a total of 9 distinct levels.

For all datasets, we exclude samples with missing values in multi-criteria ratings. Additionally, we limit our analysis to users who have rated more than 10 items, as inactive users are less pertinent in the system.. We show details of the datasets employed are given in the Table II.

TABLE II
THE STATISTICS OF DATASETS.

Datasets	#Users	#Items	#Ratings	Sparsity	Scale
TripAdvisor	1595	539	10273	98.81%	[1,5]
Yahoo!Movies	1797	1279	39489	98.28%	[1,5]
BeerAdvocate	3238	2893	88242	99.06%	[1,5]

We employ cross-validation method to evaluate the accuracy of each approach based on five independent instances over training and test set [54], [55]. The training set consists of 80% of the original data and the remaining data comprises the test set. Our purpose is predicting individual overall ratings and

Algorithm 1 Collective factor model.

Input:
$\mathbf{R}^{(0)}$: the overall rating matrix;
$\mathbf{R}^{(\alpha)}$ $(\alpha = 1, 2, ..., c)$: the multi-criteria rating matrix;
γ: the learning rate;
$Iter$: the iteration count;

Output:
$\mathbf{X}^{(0)}, \mathbf{Y}^{(0)}$: the user and item latent factor matrix $w.r.t$ the overall rating;
$\mathbf{X}^{(\alpha)}, \mathbf{Y}^{(\alpha)}$ $(\alpha = 1, 2, ..., c)$: the user and item latent factor matrix $w.r.t$ criterion α;
$b_u^{(0)}, b_i^{(0)}$: the bias of user and item $w.r.t$ the overall rating;
$b_u^{(\alpha)}, b_i^{(\alpha)}$ $(\alpha = 1, 2, ..., c)$: the bias of user/item $w.r.t$ criterion α ;
w_α $(\alpha = 1, 2, ..., c)$: the weight of predictive score on criterion α;

randomly initialize all parameters;
for *it* from 1 to *Iter* **do**
  for each $(r_{ui}^{(0)}, r_{ui}^{(1)}, r_{ui}^{(2)}, ..., r_{ui}^{(c)})$ in training set **do**
    // Computing Loss
    set $\hat{r}_{ui}^{(0)} \leftarrow$ use Eq. (6) with input $(\mathbf{x}_u^{(0)}, \mathbf{y}_i^{(0)})$;
    for α from 1 to c **do**
      set $\hat{r}_{ui}^{(\alpha)} \leftarrow$ use Eq. (8) with input $(\mathbf{x}_u^{(\alpha)}, \mathbf{y}_i^{(\alpha)})$;
    end for
    set $\hat{r}_{ui} \leftarrow$ use Eq. (9) with input $\hat{r}_{ui}^{(0)}, \hat{r}_{ui}^{(1)}, \hat{r}_{ui}^{(2)}, ..., \hat{r}_{ui}^{(c)}$;
    set $L \leftarrow$ use Eq. (10) with input of $\hat{r}_{ui},\ r_{ui}^{(0)}$;
    // Updating
    update $\mathbf{x}_u^{(0)} \leftarrow \mathbf{x}_u^{(0)} - \gamma \frac{\partial L}{\partial \mathbf{x}_u^{(0)}}$;
    update $\mathbf{y}_i^{(0)} \leftarrow \mathbf{y}_i^{(0)} - \gamma \frac{\partial L}{\partial \mathbf{y}_i^{(0)}}$;
    update $b_u^{(0)} \leftarrow b_u^{(0)} - \gamma \frac{\partial L}{\partial b_u^{(0)}}$;
    update $b_i^{(0)} \leftarrow b_i^{(0)} - \gamma \frac{\partial L}{\partial b_i^{(0)}}$;
    for α from 1 to c **do**
      update $\mathbf{x}_u^{(\alpha)} \leftarrow \mathbf{x}_u^{(\alpha)} - \gamma \frac{\partial L}{\partial \mathbf{x}_u^{(\alpha)}}$;
      update $\mathbf{y}_i^{(\alpha)} \leftarrow \mathbf{y}_i^{(\alpha)} - \gamma \frac{\partial L}{\partial \mathbf{y}_i^{(\alpha)}}$;
      update $b_u^{(\alpha)} \leftarrow b_u^{(\alpha)} - \gamma \frac{\partial L}{\partial b_u^{(\alpha)}}$
      update $b_i^{(\alpha)} \leftarrow b_i^{(\alpha)} - \gamma \frac{\partial L}{\partial b_i^{(\alpha)}}$;
      update $w_\alpha \leftarrow w_\alpha - \gamma \frac{\partial L}{\partial w_\alpha}$;
    end for
    update $\varepsilon \leftarrow \varepsilon - \gamma \frac{\partial L}{\partial \varepsilon}$;
  end for
end for

we select *Mean Absolute Error* (MAE) and *Root Mean Square Error* (RMSE) [5], [56]–[58] to evaluate the performance of all models considered:

$$MAE = \frac{1}{|E^P|} \sum_{(u,i) \in E^P} |r_{ui}^{(0)} - \hat{r}_{ui}|$$
$$RMSE = \sqrt{\frac{1}{|E^P|} \sum_{(u,i) \in E^P} (r_{ui}^{(0)} - \hat{r}_{ui})^2}. \tag{13}$$

1) Baseline methods: We compare our method with 8 baseline methods. Specifically,

- **UserKNN**. This is the standard user-based collaborative filtering method. We employ the Pearson correlation to measure similarities of user pairs. Top-100 users are selected as the target user's neighbors.
- **MultiUserKNN**. This method calculates the similarity of two users on each criterion and similarities on all criteria are averaged as the final similarity between these two users. In a similar way, the number of the nearest neighbors is also set to 100.
- **Biased matrix factorization (BMF)** [7]. BMF is the base model of our method. In this method, we only make use of the overall ratings to train parameters in the model.
- **Multilinear singular value decomposition (MSVD)** [59]. Li et al. devised MSVD to integrate explicit and implicit relations among user, item and criterion. The approximation tensor is usually obtained by reserving the largest k-model singular values.
- **Multiple linear regressions (MLR)** [60]. This method applies the multiple linear regression model to study the relationship between a user's multi-criteria ratings and his/her overall rating. We utilize the multiple regression model to predict an individual overall rating.
- **Support vector regression (SVR)** [11]. This method trained two support regression models from user- and item- side respectively and combined these two regression models to predict the overall ratings.
- **Criteria-independent contextual model (CIC)** [3]. The multi-criteria ratings are initially estimated by the context-aware recommendation algorithm and the support vector regression is applied to predict the overall ratings.
- **Deep multi-criteria collaborative filtering (DMCF)** [6]. DMCF is a two-stages based approach which firstly adopts a neural network to estimate a user's multi-criteria ratings and secondly choose another neural network to predict his/her overall rating.
- **Deep Neural Network Matrix Factorization (DNN-MF) [24].** DNN-MF is a deep learning-based method proposed for information filtering in multi-criteria recommender systems. It fuses DNN, MF, and social spider optimization (SSO) to exploit non-linear interactions between users in terms of multi-criteria attributes.
- **MCAE-FADNN [26]** is another approach for multi-criteria recommendation systems. MCAE-FADNN leverages autoencoders with dropout layers to predict missing criteria ratings. Subsequently, it builds non-linear interaction between users and items using DNN with optimized weights attained using the firefly algorithm.

2) Parameter settings: The regularization parameter λ_0 for overall ratings and λ_α for ratings on criterion α are set to 0.001 and 0.005 after trials, respectively. All parameters are initialized with a Gaussian distribution (with a mean of 0 and a standard deviation of 0.01) [61]. For those regression based methods (*MLR* and *SVR*), we employ the *BMF* method to predict multi-criteria ratings of the target item and apply those predicted scores to calculate the overall rating. For the DMCF, we simply follow the setting of the original article [6]. We set the size of each hidden layer to $(128, 64, 32, 16, 8)$ to estimate a user's multi-criteria ratings, and set the size of the hidden layers of the high-level MLP to $(64, 32, 16, 8)$ to predict the overall rating.

B. Results and analysis

1) Relevance of multi-criteria ratings: We first study the correlation between overall ratings and multi-criteria ratings

TABLE III
THE RESULT OF LINEAR REGRESSION ANALYSIS.

Dataset	*Adjusted R-squared*	*F-statistic*	*Prob(F-statistic)*
TripAdvisor	0.819	6236	< 0.0001
Yahoo!Movies	0.837	2.998×10^4	< 0.0001
BeerAdvocate	0.629	4.064×10^4	< 0.0001

before comparing our method with baseline approaches. In general, the information of the multi-criteria rating is useful for improving the predictive accuracy only if the overall rating has a high correlation with the multi-criteria rating. Following the work in [11], [60], we assume the overall rating has a linear relation with the multi-criteria rating. Therefore, we employ a linear regression analysis between the overall rating and the multi-criteria rating.

In the linear regression model, a user's overall rating is regarded as a variable depending on multi-criteria ratings. Those sub-rates are considered as independent variables. We assume the overall that rating is a linear combination of multi-criteria ratings:

$$r_{ui}^{(0)} = \sum_{\alpha=1}^{c} b_\alpha r_{ui}^{(\alpha)} + \epsilon. \tag{14}$$

We compute 3 different metrics to evaluate the correlations between ratings, namely *Adjusted R-squared*, *F-statistic* and *Prob(F-statistic)*. The *Adjusted R-squared* is considered as an unbiased estimator of *R-squared*, that evaluates the proportion of the variance in the dependent variable which is predictable from the independent variables. We use *Adjusted R-squared* to assess the goodness-of-fit for the linear regression analysis. The *F-statistic* and *Prob(F-statistic)* test the null hypothesis that all of the regression coefficients are equal to zero.

The result of the linear regression analysis is given in Table III. By looking at the values of *F-statistic* and *Prob(F-statistic)*, we can see that test results on three datasets are significant since *Prob(F-statistic)* < 0.0001. This means that individual overall ratings have high correlations with their corresponding multi-criteria ratings, and the information of the multi-criteria rating can improve the accuracy of the overall rating prediction. The test on *BeerAdvocate* dataset has the smallest *Adjusted R-squared* value (0.629), which reveals that *BeerAdvocate* users' overall ratings have the lowest correlations with multi-criteria ratings. In other words, for the *BeerAdvocate* dataset, the accuracy improvements of the overall rating prediction may be subtle.

We further study the convergence of w_α in Eq. (9) during training, presented in Figure 2. Each curve in the Figure represents the weight of the criterion and x-axis is the iterative epochs of the gradient descent method. The w_α is randomly initialized with a Gaussian distribution with a mean of zero and a standard deviation of 0.01. Consequently, the start value of w_α is around zero for different criteria. When the iteration goes over 100, w_α starts to converge. For *TripAdvisor* dataset, weights of criteria are close to each other (around 0.15). The weight of *Location* criterion is slightly lower than the weight of remaining criteria. For *Yahoo!Movies* dataset, weights of criteria varies, ranging from 0.2 to 0.35. In the *BeerAdvocate* dataset, weights of *Aroma* and *Palate* criterion are close and *Taste* criterion has the highest weight. A potential reason is that users may be more concerned about the taste when they choose beers.

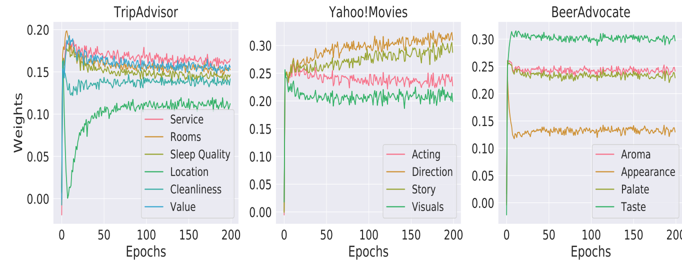

Fig. 2. The convergence of w_α.

2) The performance comparisons of different methods: We show the performance of all models considered on different models and dataset in Table IV. Among these methods, *UserKNN* and *BMF* are one-step approaches that only employ overall ratings. *MultiUserKNN* is an extension of *UserKNN* that utilizes both overall ratings and multi-criteria ratings to compute similarities between user pairs. We can see that the accuracy of *MultiUserKNN* is slightly worse than the accuracy of *UserKNN*, which means *MultiUserKNN* is not an effective way to uncover the information of multi-criteria ratings.

However, not all multi-criteria based approaches outperform the single-criterion based method *BMF*, though they exploit more information. For instance, the error of *MLR* and *SVR* is higher of *BMF*. These regression based methods initially predict multi-criteria ratings of the target item. However, this sub-process also imposes prediction error, which is amplified in the final stage and leads to poor predictions of overall ratings. Turning attention to the *MAE* metric, *CIC* achieves lower error than *BMF* on *Yahoo!Movies* dataset. However, *CIC* performs worse than *BMF* on remaining datasets. This implies that the performance of *CIC* do not generalize well on different applications. In addition, the deep learning based methods (e.g., *DMCF*) achieves worse performance than our method (CFM_{ind}). This is because that the above methods is not trained *end-to-end* [62], [63], which amplifies the error in its sub-process.

Among those methods, CFM_{ind} achieves the best performance, as it obtains the lowest error over all baselines. Although CFM_{user} and CFM_{item} take advantage of the transfer learning and share the latent vector of the user and item, they do not outperform the independent variant CFM_{ind}. When the target data is sparse, the knowledge sharing method may be helpful in improving the predictive accuracy of the target domain [30]. However, in the multi-criteria recommender system, the overall rating has the same sparsity with the multi-criteria rating. This means that sharing the user'(item') latent vector may dilute the knowledge in the domain of the overall rating, and therefore it is better to keep the latent space independent in the multi-criteria recommender systems.

Although CFM_{ind} outperforms *BMF* on three datasets, the improvements are various from different datasets. We can

TABLE IV
THE PERFORMANCE OF RECOMMENDATION APPROACHES. THE STANDARD ERROR IS PRESENTED IN THE BRACKET. BOLD VALUES INDICATE THE BEST RESULTS.

	RMSE			MAE		
	TripAdvisor	*Yahoo!Movies*	*BeerAdvocate*	*TripAdvisor*	*Yahoo!Movies*	*BeerAdvocate*
UserKNN	1.2159(0.0239)	1.2329(0.0696)	0.8444(0.0150)	0.9458(0.0252)	0.9260(0.0716)	0.6559(0.0169)
MultiUserKNN	1.2146(0.0238)	1.2396(0.0715)	0.8441(0.0161)	0.9458(0.0255)	0.9319(0.0729)	0.6572(0.0177)
BMF	0.6820(0.0088)	0.8646(0.0061)	0.5858(0.0009)	0.4032(0.0023)	0.6289(0.0032)	0.4394(0.0001)
MSVD	0.9505(0.0596)	0.8738(0.0046)	0.5960(0.0006)	0.6387(0.0109)	0.6332(0.0030)	0.4473(0.0039)
MLR	0.7475(0.0081)	0.8664(0.0060)	0.5929(0.0002)	0.5255(0.0009)	0.6326(0.0066)	0.4442(0.0006)
SVR	0.7465(0.0086)	0.8671(0.0058)	0.5993(0.0021)	0.5109(0.0038)	0.6248(0.0063)	0.4470(0.0051)
CIC	0.6836(0.0140)	0.8782(0.0185)	0.5914(0.0070)	0.4055(0.0054)	0.6200(0.0055)	0.4429(0.0053)
DMCF	0.8289(0.0101)	0.9139(0.0078)	0.6240(0.0098)	0.5819(0.0028)	0.7012(0.0017)	0.4698(0.0058)
DNN-MF	0.7606(0.0032)	0.8606(0.0008)	0.6077(0.0058)	0.5334(0.0154)	0.6178(0.0019)	0.4483(0.0028)
MCAE-FADNN	0.8301(0.0006)	0.8793(0.0002)	0.6240(0.0098)	0.6031(0.0153)	0.6277(0.0045)	0.4698(0.0006)
CFM_{user}	0.6492(0.0129)	0.8802(0.0095)	0.5904(0.0019)	0.3965(0.0013)	0.6184(0.0080)	0.4403(0.0001)
CFM_{item}	0.6549(0.0095)	0.8869(0.0035)	0.5904(0.0017)	0.3898(0.0021)	0.6145(0.0046)	0.4408(0.0008)
CFM_{ind}	**0.6117(0.0011)**	**0.8514(0.0063)**	**0.5833(0.0003)**	**0.3522(0.0065)**	**0.6042(0.0020)**	**0.4360(0.0004)**

see that CFM_{ind} perform better than *BMF* on *BeerAdvocate* dataset. This is because users' overall ratings in *BeerAdvocate* have the low correlations with their multi-criteria ratings. According to Eq.(9), our method combines contributions of overall ratings and multi-criteria ratings in a linear manner, which leads to a modest performance of our method on *BeerAdvocate* dataset.

Overall, our proposed CFMs obtain the best performance on all datasets, by achieving up to 10.52% and 13.14% lower RMSE and MAE than the state-of-the-art approach CIC.

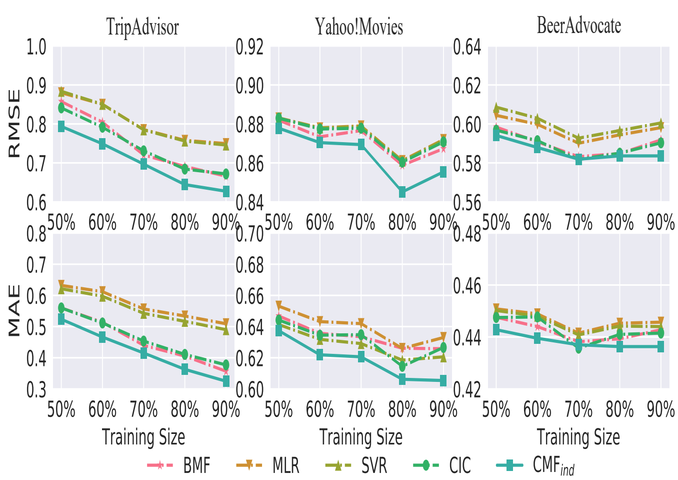

Fig. 3. The performance of methods $w.r.t.$ different training sizes.

We then do a sign test on the results to show that the improvements are indeed significant. In the sign test, a z^* score is computed by $z^* = \frac{n_A - 0.5(n_A+n_B)}{\sqrt{(n_A+n_B)/4}}$, where n_A denotes the number of users that CFM_{ind} is better than the baseline method. n_B is the number of users that the baseline method is superior to our method. Our method and the baseline method are significantly different when $|z^*| > 1.96$. Table V shows the results of the significant analysis. We can see that our method (CFM_{ind}) outperforms baseline methods for most users. The result of z^* score also indicates that our method and baseline methods are significantly different ($|z^*| > 1.96$). However, one exception exists, as on *BeerAdvocate* dataset, the difference between our method and *CIC* is not significant. This is because users' overall ratings in this dataset have relatively low correlations with their multi-criteria ratings.

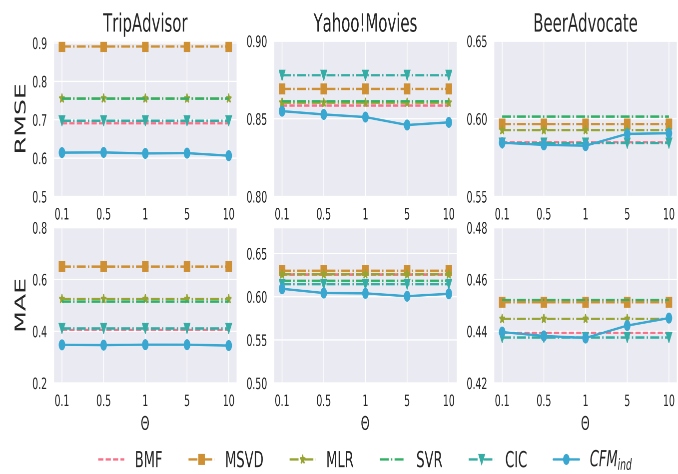

Fig. 4. The performance of our method with different values of Θ.

C. The impact of sparsity and hyper-parameter

We study the performance of the model when the sparsity of the dataset varies, as shown in Figure 3. The x-axis is the proportion of overall ratings in the training set to the total number of all overall ratings, where the y-axis is the error metric. In general, CFM_{ind} outperforms baseline approaches when the training size ranges from 50% to 90%, which shows the superiority of our method. On the *BeerAdvocate* dataset, CFM_{ind}'s *MAE* is slightly higher than *MAE* of *CIC* when the training size is 70%.

To evaluate the proposed model against state-of-the-art baselines in the scenario where users have few ratings, we conducted experiments on the original dataset, which includes

TABLE V
THE SIGNIFICANT TEST OF ALGORITHMS.

	TripAdvisor		*Yahoo!Movies*		*BeerAdvocate*	
$n_A/(n_A+n_B)*100\%$						
	RMSE	MAE	RMSE	MAE	RMSE	MAE
UserKNN	79.88%	77.42%	85.81%	86.83%	80.95%	78.77%
MultiUserKNN	79.82%	77.67%	85.62%	86.46%	79.94%	77.99%
BMF	56.29%	59.03%	55.94%	55.09%	52.70%	53.71%
MSVD	55.99%	59.85%	78.80%	80.46%	57.22%	56.18%
MLR	57.06%	63.09%	68.96%	71.93%	54.00%	55.04%
SVR	54.93%	60.29%	68.02%	71.05%	57.90%	55.23%
CIC	53.52%	56.70%	55.28%	55.84%	50.26%	51.24%
Average	62.50%	64.86%	71.20%	72.52%	61.85%	61.16%
z^*score						
UserKNN	23.51	24.18	24.12	22.14	34.33	31.91
MultiUserKNN	23.39	23.94	24.08	22.34	33.21	31.05
BMF	3.87	3.32	5.04	7.24	2.99	4.11
MSVD	18.93	20.02	4.79	7.89	8.01	6.85
MLR	12.44	14.39	5.73	10.61	4.44	5.59
SVR	11.72	13.69	4.00	8.34	8.76	5.81
CIC	3.45	3.82	2.82	5.37	0.29	1.37
Average	13.90	14.77	10.08	11.99	13.15	12.38

TABLE VI
THE PERFORMANCE OF METHODS ON UNFILTERED DATASETS.

	TripAdvisor		*BeerAdvocate*	
	RMSE	MAE	RMSE	MAE
BMF	0.7709	0.4477	0.9811	0.7266
MLR	0.7708	0.5373	0.7916	0.6018
SVR	0.7817	0.5470	1.2216	1.0977
CIC	0.7647	0.3989	0.7836	0.5916
DNN-MF	0.7956	0.5576	0.7947	0.5952
MCAE-FADNN	0.7688	0.5283	0.7870	0.5927
CFM_{ind}	**0.6852**	**0.3239**	**0.6759**	**0.5119**

users with less than 10 ratings. We present the results in Table VI and excluded Yahoo! Movies dataset since all users in this dataset had more than 10 ratings. Our experimental results demonstrate that the proposed CFM model outperforms the baseline method even when trained on such sparse and noisy datasets. Specifically, the rating prediction task is more challenging in the unfiltered datasets due to their sparsity and noise. Nonetheless, the CFM model has achieved a greater relative improvement, indicating its superior adaptability to the sparse data scenario. This can be attributed to its ability to capture information from multiple aspects, including both overall and criteria ratings.

TABLE VII
THE TRAINING TIME OF METHODS. THE TRAINING TIME IS MEASURED BY SECONDS.

	TripAdvisor	*Yahoo!Movies*	*BeerAdvocate*
BMF	48.22	193.67	435.21
MSVD	46.27	178.52	423.87
MLR	289.32	774.68	1741.12
SVR	291.45	777.11	1750.99
CIC	344.80	983.69	2222.41
CFM$_{ind}$	83.31	363.12	768.56

Our method adopts a linear method to combine the loss function of the overall rating and the multi-criteria rating (see Eq.(10)) and employs a hyper-parameter (Θ_α) to control contributions of multi-criteria ratings. We then study the performance of our method with different values of Θ_α. For all criteria, we use the same value of Θ (i.e. $\Theta_1 = \Theta_2 = ... = \Theta_c$). The result is shown in Figure 4. We can see that our method achieves the best predictive accuracy when Θ is around 1 for *TripAdvisor* and *BeerAdvocate* datasets. This indicates that individual multi-criteria rating has almost equal contribution with the corresponding overall rating. For the *Yahoo!Movies* dataset, the optimal Θ is around 5.

D. The complexity analysis

Finally, we evaluate the training time of different models. All these methods are implemented by an open-source machine learning framework TensorFlow. We use the same machine (Intel I7 6800K CPU) to perform the training process. The results are shown in Table VII where the training time is measured by seconds. Among these methods, *MSVD* requires the least training time. Although *MSVD* is faster than the iterative optimization methods (i.e. SGD), it is difficult to apply such method to the large-scale dataset since the time complexity of computing the singular value is cubic. Although our method consumes more training time than *BMF* and *MSVD*, the training time of our method is far lower than of *MLR*, *SVR* and *CIC*, which requires considerable amount of time to predict multi-criteria ratings of the target item. Those results show that our proposed en-to-end method can speed up training compared to other two-stages based approaches.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose an end-to-end collective factor model (CFM) for multi-criteria recommender systems. Our method integrates the loss functions of overall ratings and multi-criteria ratings linearly, allowing us to train the collective factor model using both types of ratings. Our theoretical analysis shows that the proposed method can transfer the knowledge of each criterion to the domain of overall ratings, thereby reducing the dependency on sub-processes. Our experiments on three benchmark datasets demonstrate that our method outperforms eight different baselines, achieving up to 10.52% and 13.14% lower RMSE and MAE, respectively, compared to the state-of-the-art approach of CIC. Additionally, our approach is more efficient than traditional two-stage approaches, resulting in considerable time savings during training.

Although the proposed CFM predicts ratings based on the BMF model and models the overall and sub-ratings with linear weights, making it relatively simple, general, and easy to train, it overlooks non-linear relationships. This may limit the prediction performance of the proposed CFM. Future work will focus on exploring deep learning-based structures to improve the model's ability to capture non-linear relationships, including the relationships between users and items, and between overall ratings and sub-ratings.

REFERENCES

[1] G. Adomavicius and Y. Kwon, "New recommendation techniques for multicriteria rating systems," *IEEE Intelligent Systems*, vol. 22, no. 3, pp. 48–55, May 2007.
[2] M. Nilashi, D. Jannach, O. bin Ibrahim, and N. Ithnin, "Clustering- and regression-based multi-criteria collaborative filtering with incremental updates," *Information Sciences*, vol. 293, pp. 235 – 250, 2015.

[3] Y. Zheng, "Criteria chains: A novel multi-criteria recommendation approach," in *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, ser. IUI '17. ACM, 2017, pp. 29–33.

[4] A. Mikeli, D. Sotiros, D. Apostolou, and D. Despotis, "A multi-criteria recommender system incorporating intensity of preferences," in *4th International Conference on Information, Intelligence, Systems and Applications*, ser. IISA '13. IEEE, July 2013, pp. 1–6.

[5] Q. Shambour, "A deep learning based algorithm for multi-criteria recommender systems," *Knowledge-Based Systems*, p. 106545, 2021.

[6] N. Nassar, A. Jafar, and Y. Rahhal, "A novel deep multi-criteria collaborative filtering model for recommendation system," *Knowledge-Based Systems*, vol. 187, p. 104811, 2020.

[7] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proceedings of KDD cup and workshop*, ser. KDDCup '07, vol. 2007, 2007, pp. 5–8.

[8] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '08. ACM, 2008, pp. 650–658.

[9] M. Nilashi, O. bin Ibrahim, and N. Ithnin, "Hybrid recommendation approaches for multi-criteria collaborative filtering," *Expert Systems with Applications*, vol. 41, no. 8, pp. 3879 – 3900, 2014.

[10] N. R. Kermany and S. H. Alizadeh, "A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques," *Electronic Commerce Research and Applications*, vol. 21, pp. 50 – 64, 2017.

[11] D. Jannach, Z. Karakaya, and F. Gedikli, "Accuracy improvements for multi-criteria recommender systems," in *Proceedings of the 13th ACM Conference on Electronic Commerce*, ser. EC '12. ACM, 2012, pp. 674–689.

[12] P. Li and A. Tuzhilin, "Latent multi-criteria ratings for recommendations," in *Proceedings of the 13th ACM Conference on Recommender Systems*, ser. RecSys '19. ACM, 2019, pp. 428–431.

[13] J. McAuley, J. Leskovec, and D. Jurafsky, "Learning attitudes and attributes from multi-aspect reviews," in *2012 IEEE 12th International Conference on Data Mining*, ser. ICDM '12. IEEE, Dec 2012, pp. 1020–1025.

[14] K. Bauman, B. Liu, and A. Tuzhilin, "Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '17. ACM, 2017, pp. 717–725.

[15] N. Sahoo, R. Krishnan, G. Duncan, and J. P. Callan, "Collaborative filtering with multi-component rating for recommender systems," in *Proceedings of the 16th workshop on information technologies and systems*, ser. ITS '16 Workshop, 2006, pp. 1–6.

[16] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering," in *Proceedings of the 4th ACM Conference on Recommender Systems*, ser. RecSys '10. ACM, 2010, pp. 79–86.

[17] M. Nilashi, O. bin Ibrahim, and N. Ithnin, "Multi-criteria collaborative filtering with high accuracy using higher order singular value decomposition and neuro-fuzzy system," *Knowledge-Based Systems*, vol. 60, pp. 82 – 101, 2014.

[18] M. Pozo, R. Chiky, and E. Métais, "Enhancing collaborative filtering using implicit relations in data," in *Transactions on Computational Collective Intelligence XXII*, ser. TCCI '16, 2016, pp. 125–146.

[19] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Tensors for data mining and data fusion: Models, applications, and scalable algorithms," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 2, pp. 16:1–16:44, Oct. 2016.

[20] H. Morise, S. Oyama, and M. Kurihara, "Bayesian probabilistic tensor factorization for recommendation and rating aggregation with multicriteria evaluation data," *Expert Systems with Applications*, vol. 131, pp. 1–8, 2019.

[21] M. Hong and J. J. Jung, "Sentiment aware tensor model for multi-criteria recommendation," *Applied Intelligence*, pp. 1–20, 2022.

[22] Q.-H. Le, T. N. Mau, R. Tansuchat, and V.-N. Huynh, "A multi-criteria collaborative filtering approach using deep learning and dempster-shafer theory for hotel recommendations," *IEEE Access*, vol. 10, pp. 37 281–37 293, 2022.

[23] A. R. Kalantarnezhad and J. Hamidzadeh, "Mcrs-sae: multi-criteria recommender system based on sparse autoencoder," in *2022 12th International Conference on Computer and Knowledge Engineering (ICCKE)*. IEEE, 2022, pp. 117–122.

[24] B. B. Sinha and R. Dhanalakshmi, "Dnn-mf: Deep neural network matrix factorization approach for filtering information in multi-criteria recommender systems," *Neural Computing and Applications*, pp. 1–15, 2022.

[25] Q.-H. L. Son-Lam VU, "A deep learning based approach for context-aware multi-criteria recommender systems," *Computer Systems Science and Engineering*, vol. 44, no. 1, pp. 471–483, 2023.

[26] G. Spoorthy and S. Sanjeevi, "Multi-criteria–recommendations using autoencoder and deep neural networks with weight optimization using firefly algorithm," *International Journal of Engineering*, vol. 36, no. 1, pp. 130–138, 2023.

[27] C. V. M. Krishna, G. A. Rao, and S. Anuradha, "Analysing the impact of contextual segments on the overall rating in multi-criteria recommender systems," *Journal of Big Data*, vol. 10, no. 1, pp. 1–35, 2023.

[28] R. Shrivastava, D. S. Sisodia, and N. K. Nagwani, "Deep neural network-based multi-stakeholder recommendation system exploiting multi-criteria ratings for preference learning," *Expert Systems with Applications*, vol. 213, p. 119071, 2023.

[29] J. Chen, G. Fan, Z. Gong, X. Li, V. C. Leung, M. Wang, and M. Yang, "A topic-aware graph-based neural network for user interest summarization and item recommendation in social media," in *Database Systems for Advanced Applications: 28th International Conference*, ser. DASFAA'23, 2023, pp. 537–546.

[30] W. Pan and Q. Yang, "Transfer learning in heterogeneous collaborative filtering domains," *Artificial Intelligence*, vol. 197, pp. 39 – 55, 2013.

[31] L. Chen, W. Zeng, and Q. Yuan, "A unified framework for recommending items, groups and friends in social media environment via mutual resource fusion," *Expert Systems with Applications*, vol. 40, no. 8, pp. 2889–2903, 2013.

[32] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: Social recommendation using probabilistic matrix factorization," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, ser. CIKM '08. ACM, 2008, pp. 931–940.

[33] K. Zhang, X. Liu, W. Wang, and J. Li, "Multi-criteria recommender system based on social relationships and criteria preferences," *Expert Systems with Applications*, vol. 176, p. 114868, 2021.

[34] J. Chen, Z. Gong, Y. Li, H. Zhang, H. Yu, J. Zhu, G. Fan, X.-M. Wu, and K. Wu, "Meta-path based neighbors for behavioral target generalization in sequential recommendation," *IEEE Transactions on Network Science and Engineering*, 2022.

[35] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang, "Collaborative location and activity recommendations with gps history data," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. ACM, 2010, pp. 1029–1038.

[36] G. Zhao, X. Qian, and C. Kang, "Service rating prediction by exploring social mobile users' geographic locations," *IEEE Transactions on Big Data*, vol. 3, no. 1, pp. 67–78, March 2017.

[37] Z. Zhang, C. Zou, R. Ding, and Z. Chen, "Vcg: Exploiting visual contents and geographical influence for point-of-interest recommendation," *Neurocomputing*, 2019.

[38] M. Saveski and A. Mantrach, "Item cold-start recommendations: Learning local collective embeddings," in *Proceedings of the 8th ACM Conference on Recommender Systems*, ser. RecSys '14, 2014, pp. 89–96.

[39] Y. Fang and L. Si, "Matrix co-factorization for recommendation with rich side information and implicit feedback," in *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, ser. HetRec '11. ACM, 2011, pp. 65–69.

[40] J. Liu, Y. Jiang, Z. Li, Z.-H. Zhou, H. Lu *et al.*, "Partially shared latent factor learning with multiview data," *IEEE Transactions on Neural Networks and Learning Systems*, no. 6, pp. 1233–1246, June 2015.

[41] N. N. Liu, E. W. Xiang, M. Zhao, and Q. Yang, "Unifying explicit and implicit feedback for collaborative filtering," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, ser. CIKM '10. ACM, 2010, pp. 1445–1448.

[42] Q. Yuan, L. Chen, and S. Zhao, "Factorization vs. regularization: Fusing heterogeneous social relationships in top-n recommendation," in *Proceedings of the Fifth ACM Conference on Recommender Systems*, ser. RecSys '11. ACM, 2011, pp. 245–252.

[43] S.-H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha, "Like like alike: Joint friendship and interest propagation in social networks," in *Proceedings of the 20th International Conference on World Wide Web*, ser. WWW '11. ACM, 2011, pp. 537–546.

[44] Z. Zhao, Z. Cheng, L. Hong, and E. H. Chi, "Improving user topic interest profiles by behavior factorization," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15, 2015, pp. 1406–1416.

[45] G. Hu, X. Dai, F. Qiu, R. Xia, T. Li, S. Huang, and J. Chen, "Collaborative filtering with topic and social latent factors incorporating

implicit feedback," *ACM Transactions on Knowledge Discovery from Data*, vol. 12, no. 2, pp. 23:1–23:30, Jan. 2018.

[46] V. W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang, "Collaborative filtering meets mobile recommendation: A user-centered approach," in *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, ser. AAAI '10. AAAI Press, 2010, pp. 236–241.

[47] G. Fan, C. Zhang, J. Chen, B. Li, Z. Xu, Y. Li, L. Peng, and Z. Gong, "Field-aware variational autoencoders for billion-scale user representation learning," in *2022 IEEE 38th International Conference on Data Engineering*, ser. ICDE'22. IEEE, 2022, pp. 3413–3425.

[48] G. Guo, J. Zhang, Z. Sun, and N. Yorke-Smith, "Librec: A java library for recommender systems," in *Proceedings of the 23rd Conference on User Modelling, Adaptation and Personalization*, ser. UMAP '15 Workshop, vol. 4, 2015, pp. 1–4.

[49] G. Fan, B. Geng, J. Tao, K. Wang, C. Fan, and W. Zeng, "PPPNE: Personalized proximity preserved network embedding," *Neurocomputing*, vol. 472, pp. 103–112, 2022.

[50] G. Fan, C. Zhang, J. Chen, and K. Wang, "MV-HAN: A hybrid attentive networks based multi-view learning model for large-scale contents recommendation," in *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE'22, 2022.

[51] H. Wang, Y. Lu, and C. Zhai, "Latent aspect rating analysis without aspect keyword supervision," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '11. ACM, 2011, pp. 618–626.

[52] D. Jannach, M. Zanker, and M. Fuchs, "Leveraging multi-criteria customer feedback for satisfaction analysis and improved recommendations," *Information Technology & Tourism*, vol. 14, no. 2, pp. 119–149, Jul 2014.

[53] Y. Zheng and D. Wang, "Multi-criteria ranking: Next generation of multi-criteria recommendation framework," *IEEE Access*, vol. 10, pp. 90 715–90 725, 2022.

[54] W. Zeng, M. Fang, J. Shao, and M. Shang, "Uncovering the essential links in online commercial networks," *Scientific Reports*, vol. 6, p. 34292, Sep 2016.

[55] J. J. McAuley and J. Leskovec, "From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews," in *Proceedings of the 22nd International Conference on World Wide Web*, ser. WWW '13. ACM, 2013, pp. 897–908.

[56] G. Fan, C. Zhang, J. Chen, and K. Wu, "Predicting ratings in multi-criteria recommender systems via a collective factor model," in *DeMaL @ The Web Conference*, 2021.

[57] C. Zhang, M. Fiore, C. Ziemlicki, and P. Patras, "Microscope: mobile service traffic decomposition for network slicing as a service," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020, pp. 1–14.

[58] C. Zhang, K. Wang, H. Chen, G. Fan, Y. Li, L. Wu, and B. Zheng, "Quickskill: Novice skill estimation in online multiplayer games," in *Proceedings of the 31st ACM Conference on Information and Knowledge Management*, ser. CIKM '22. ACM, 2022, pp. 1–10.

[59] Q. Li, C. Wang, and G. Geng, "Improving personalized services in mobile commerce by a novel multicriteria rating approach," in *Proceedings of the 17th International Conference on World Wide Web*, ser. WWW '08. ACM, 2008, pp. 1235–1236.

[60] M. Fuchs and M. Zanker, "Multi-criteria ratings for recommender systems: an empirical analysis in the tourism domain," in *International Conference on Electronic Commerce and Web Technologies*. Springer, 2012, pp. 100–111.

[61] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, ser. NIPS'07, 2007, pp. 1257–1264.

[62] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, ser. NIPS'15, 2015.

[63] H. Huang, Q. Zhang, X. Huang, H. Huang, Q. Zhang, and X. Huang, "Mention recommendation for twitter with end-to-end memory network," in *IJCAI*, 2017, pp. 1872–1878.