Exploring Hierarchical Structures for Recommender Systems

Suhang Wang, Jiliang Tang, Yilin Wang, and Huan Liu, *Fellow, IEEE*

Abstract—Items in real-world recommender systems exhibit certain hierarchical structures. Similarly, user preferences also present hierarchical structures. Recent studies show that incorporating the hierarchy of items or user preferences can improve the performance of recommender systems. However, hierarchical structures are often not explicitly available, especially those of user preferences. Thus, there's a gap between the importance of hierarchies and their availability. In this paper, we investigate the problem of exploring the implicit hierarchical structures for recommender systems when they are not explicitly available. We propose a novel recommendation framework to bridge the gap, which enables us to explore the implicit hierarchies of users and items simultaneously. We then extend the framework to integrate explicit hierarchies when they are available, which gives a unified framework for both explicit and implicit hierarchical structures. Experimental results on real-world datasets demonstrate the effectiveness of the proposed framework by incorporating implicit and explicit structures.

Index Terms—Recommender system, implicit hierarchical structures, explicit hierarchical structures, deep nonnegative factorization

1 Introduction

Recommender systems [1] intend to provide users with information of potential interest based on their demographic profiles and historical data. Collaborative Filtering (CF), which only requires past user ratings to predict unknown ratings, has attracted more and more attention [2], [3], [4]. Collaborative Filtering can be roughly categorized into memory-based [5], [6], [7] and model-based methods [2], [8], [9]. Memory-based methods mainly use the neighborhood information of users or items in the user-item rating matrix while model-based methods usually assume that an underlying model governs the way users rate and in general, and model-based methods have better performance than memory-based methods. Despite the success of various model-based methods [2], [10], matrix factorization (MF) based model has become one of the most popular methods due to its good performance and efficiency in handling large datasets [8], [9], [11], [12], [13].

Items in real-world recommender systems could exhibit certain hierarchical structures. For example, Figs. 1a and 1b are two snapshots from Netflix DVD rental page.[1] In the figure, movies are classified into a hierarchical structure as genre→subgenre→detailed-category. For example, the movie *Schindler's List* first falls into the genre *Faith Spirituality*, under which it belongs to sub-genre *Faith & Spirituality Feature Films* and is further categorized as *Inspirational Stories* (see the hierarchical structure shown in Fig. 1a). Similarly, Fig. 1c shows an Antiques & Collectibles category from half.com.[2] We can also observe hierarchical structures, i.e., category→sub-category. For example, the book *Make Your Own Working Paper Clock* belongs to *Clocks & Watches*, which is a sub-category of *Antiques & Collections*. In addition to hierarchical structures of items, users' preferences also present hierarchical structures, which have been widely used in the research of decision making [14]. For example, a user may generally prefer movies in *Faith Spirituality*, and more specifically, he/she watches movies under the subcategory of *Inspirational Stories*. Similarly, an antique clock collector may be interested in *Clocks & Watches* subcategory under the *Antiques & Collections* category. Items in the same hierarchical layer are likely to share similar properties, hence they are likely to receive similar rating scores. Similarly, users in the same hierarchical layer are likely to share similar preferences, thus they are likely to rate certain items similarly [15], [16]. Therefore, recent recommender systems exploit explicit hierarchies of items or users [15], [16], [17], [18]. However, explicit hierarchies are often unavailable, especially those of user preferences.

The gap between the importance of hierarchical structures and their unavailability motivates us to study implicit hierarchical structures of users and items for recommendation. In particular, we investigate the following two challenges: (1) how to capture implicit hierarchical structures of users and items simultaneously when these structures are explicitly unavailable? and (2) how to model them mathematically for recommendation? In our attempt to address

1. Snapshots are from http://dvd.netflix.com/AllGenresList

2. Snapshot is from http://books.products.half.ebay.com/antiques-collectibles_W0QQcZ4QQcatZ218176

- *S. Wang, Y. Wang, and H. Liu are with the Department of Computer Science and Engineering, Arizona State University, Tempe, AZ, 85281. E-mail: {Suhang.Wang, Yilin.Wang, Huan.Liu}@asu.edu.*
- *J. Tang is with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824. E-mail: tangjili@msu.edu.*

Manuscript received 3 May 2017; revised 13 Nov. 2017; accepted 23 Dec. 2017. Date of publication 4 Jan. 2018; date of current version 27 Apr. 2018. (Corresponding author: Suhang Wang.)
Recommended for acceptance by D. Cai.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2018.2789443

(a) Faith & Spirituality (b) Music & Musicals (c) Books

Fig. 1. (a) and (b) are Netflix movie hierarchical structure and (c) is half. com book hierarchical structure.

these two challenges, we propose a novel framework IHSR, which captures *I*mplicit *H*ierarchical *S*tructures of users and items based on the user-item matrix and integrate them into a coherent model. We further extend the framework to capture explicit hierarchical structures when the structures are explicitly available, which enables a unified framework HSR for both explicit and implicit *H*ierarchical *S*tructures. The major contributions of this paper are summarized next:

- We provide a principled approach to model implicit and explicit hierarchical structures of users and items simultaneously based on the user-item matrix;
- We propose a novel recommendation framework IHSR which enables us to capture implicit hierarchical structures of users and items when these structures are not explicitly available;
- We extend the proposed framework to capture explicit hierarchical structures when they are available, which results in a unified framework HSR that is able to exploit both implicit and explicit hierarchical structures for recommendation; and
- We conduct extensive experiments on four real-world recommendation datasets to demonstrate the effectiveness of the proposed framework.

The rest of the paper is organized as follows. In Section 2, we review related work. In Section 3, we introduce IHSR that explores implicit HS. In Section 4, we introduce HSR which exploits explicit HS. In Section 5, we present a method to solve the optimization problem of HSR/IHSR along with the convergence and time complexity analysis. In Section 6, we show empirical evaluation with discussion. In Section 7, we present the conclusion and future work.

2 Related Work

In this section, we will briefly review related works on recommender systems and hierarchical structures for recommendation.

2.1 Recommender Systems

Recommender systems [1] play an important role in helping online users find relevant information based on their demographic profiles and historical data. Collaborative Filtering (CF), which only requires past user ratings to predict unknown ratings, has attracted more and more attention [2], [3], [4], [19], [20], [21], [22]. Generally, collaborative filtering can be classified into two categorizes—(1) memory-based methods [5], [6], [7], which mainly use the neighborhood information of users or items in the user-item rating matrix for recommendation; and (2) model-based methods, which usually assume that an underlying model governs the way users rate [2], [8], [9]. In general, model-based methods have better performance than memory-based methods. Despite the success of various model-based methods [2], [10], matrix factorization (MF) based model has become one of the most popular methods due to its good performance and efficiency in handling large datasets [8], [9], [11], [12], [13]. The essential idea of MF based models is to decompose the user-item rating matrix into user latent feature matrix and item latent feature matrix such that user latent feature matrix captures user preferences while item latent feature matrix captures item properties.

One of the major challenges of collaborative filtering is the data sparsity problem, i.e., only a small subset of products are rated by a user and the ratings of the user to the majority of the products are unknown. For example, the density of available ratings in commercial recommender systems is often less than 1 percent [23]. The data sparsity problem degrades the performance of recommender systems. One approach to alleviate data sparsity is graph-based recommender systems [24], where the interactions or similarities between the users and items are encoded into edges such that random walks and other graph mining algorithms can be used to fully mine user preferences based on the graph. Another approach for alleviating data sparsity is top-K recommendations [25], [26], [27]. Instead of focusing on predicting the ratings, top-K recommender systems focus on the top-K ranking. Therefore, may approaches tries to learn latent representations that can keep the ranking. For example, Rendle et al. [25] proposed a Bayesian personalized ranking for top-K recommendation.

Incorporating auxiliary information into collaborative filtering is another popular and effective approach to alleviating the data sparsity problem [20], [28], [29], [30], [31], [32]. Various auxiliary information has been exploited to guide the learning process of collaborative filtering, such as social relationships [28], [33], review contents [31], [34] and temporal signals [4], [29]. For example, Ma et al. [28] incorporated social networks to matrix factorization with the assumption that friends have similar interests and significantly increases the recommendation performance. Tang et al. [30] investigated the distrust/foe relationships in social networks for recommendation and demonstrated that distrust/foe relationships have added value in addition to trust/friend relationships. Review contents are also popularly used for improving recommendation performance. Almahair et al. [34] used LSTM to infer user preferences from review texts for collaborative filtering. Recently, hierarchical structures, as a new source of auxiliary information, are attracting more and more attentions for recommender systems [15], [16], [17], [18], [26], [35], which will be discussed in next section.

2.2 Hierarchical Structures for Recommendation

Hierarchical structures of items are very pervasive in real-world. For example, as shown in Fig. 1, musics are usually first categorized into genres such as "Classical Music",

"Country & Western/Folk" and "Jazz ", which are further categorized into sub-genres, e.g., the genre "Classical Music" has sub-genres "Classical Choral Music" and "Opera & Operetta". Similarly, books are categorized into different categories by topics and sub-topics. In addition to item hierarchical structures, user preferences can also exhibit hierarchical structures. Obviously, items in the same genre or sub-genre of the hierarchical structure share some properties and thus are likely to receive similar ratings, which has been demonstrated helpful to improve the performance of recommender systems [15], [16], [18], [26]. Maleszka et al. [16] studied hierarchical structures of user profiles for recommender systems. Nikolakopoulos et al. [26] exploited the intrinsic hierarchical structure of the itemspace to tackle the data sparsity problem. Yang et al. [18] proposed a recursive matrix factorization method which uses hierarchies as a regularizer. Recent study also suggests that relationship among sibling nodes of hierarchies could also be useful [36]. However, the aforementioned methods assume that hierarchical structures are explicitly available while in real-world, explicit hierarchical structures are often unavailable, especially those of users preferences. We call the hierarchical structure that cannot be explicitly obtained as implicit hierarchical structures, meaning that we know the existence of the hierarchical structures but we don't have the explicit structure in hand. One example is hierarchical structures of user preferences, whose preference hierarchy is difficult to obtain. None of the aforementioned paper studies *implicit* hierarchical structures; while implicit hierarchical structures have potential to improve recommendation performance.

Therefore, we study the novel problem of exploring implicit hierarchical structures for recommender systems, which is inherently different from existing work. In particular, we propose a novel framework which can incorporate implicit hierarchical structures of users and items for recommendation. We further extend the framework to capture explicit hierarchical structures when the structures are explicitly available, which gives us a unified and flexible recommendation framework that can exploit both implicit and explicit hierarchical structures. A close structure to hierarchy is ontology [37], which is very precise and can improve the performance of recommender systems. However, it is hard/slow to use; while hierarchies are easier to be adopted for recommender systems.

3 The Proposed Framework for Implicit Hierarchical Structures

Throughout the paper, matrices are written as boldface capital letters such as $\mathbf{A}$ and $\mathbf{B}_i$. For an arbitrary matrix $\mathbf{M}$, $\mathbf{M}(i,j)$ denotes the (i,j)th entry of $\mathbf{M}$. $||\mathbf{M}||_F$ is the Frobenius norm of $\mathbf{M}$ and $Tr(\mathbf{M})$ is the trace norm of $\mathbf{M}$ if $\mathbf{M}$ is a square matrix. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ be the set of n users and $\mathcal{V} = \{v_1, v_2, \ldots, v_m\}$ be the set of m items. We use $\mathbf{X} \in \mathbb{R}^{n \times m}$ to denote the user-item rating matrix where $\mathbf{X}(i,j)$ is the rating score from u_i to v_j if u_i rates v_j, otherwise $\mathbf{X}(i,j) = 0$. For IHSR, we do not assume the availability of hierarchical structures of users and items, hence the input of the studied problem is only the user-item rating matrix $\mathbf{X}$, which is the same as that of traditional recommender systems. Before going into details about how to model implicit hierarchical structures of users and items, we would like to first introduce the basic model of the proposed framework.

3.1 The Basic Model

In this work, we choose weighted nonnegative matrix factorization (WNMF) as the basic model of the proposed framework, which is one of the most popular models to build recommender systems and has been proven to be effective in handling large and sparse datasets [3]. WNMF decomposes the rating matrix into two nonnegative low rank matrices $\mathbf{U} \in \mathbb{R}^{n \times d}$ and $\mathbf{V} \in \mathbb{R}^{d \times m}$, where $\mathbf{U}$ is the user preference matrix with $\mathbf{U}(i,:)$ being the preference vector of u_i, and $\mathbf{V}$ is the item characteristic matrix with $\mathbf{V}(:,j)$ being the characteristic vector of v_j. Then a rating score from u_i to v_j is modeled as $\mathbf{X}(i,j) = \mathbf{U}(i,:)\mathbf{V}(:,j)$ by WNMF. $\mathbf{U}$ and $\mathbf{V}$ can be learned by solving the following optimization problem:

$$\min_{\mathbf{U} \geq \mathbf{0}, \mathbf{V} \geq \mathbf{0}} \|\mathbf{W} \odot (\mathbf{X} - \mathbf{U}\mathbf{V})\|_F^2 + \beta(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) \tag{1}$$

where $\odot$ denotes Hadamard product. $\mathbf{W}(i,j) = 1$ if u_i rates v_j, and $\mathbf{W}(i,j) = 0$ otherwise.

3.2 Modeling Implicit Hierarchical Structures

In weighted nonnegative matrix factorization, the user preference matrix $\mathbf{U}$ and the item characteristic matrix $\mathbf{V}$ can indicate implicit flat structures of users and items respectively, which have been widely used to identify communities of users [38] and clusters of items [39]. Since both $\mathbf{U}$ and $\mathbf{V}$ are nonnegative, we can further perform nonnegative matrix factorization on them, which may pave the way to model implicit hierarchical structures of users and items for recommendation. In this section, we first give details about how to model implicit hierarchical structures based on weighted nonnegative matrix factorization, and then introduce the proposed framework IHSR.

The item characteristic matrix $\mathbf{V} \in \mathbb{R}^{d \times m}$ gives the d-dimensional feature representation of m items. Since $\mathbf{V}$ is non-negative, we can further decompose $\mathbf{V}$ into two nonnegative matrices $\mathbf{V}_1 \in \mathbb{R}^{m_1 \times m}$ and $\tilde{\mathbf{V}}_2 \in \mathbb{R}^{d \times m_1}$ to get a 2-layer implicit hierarchical structure of items as shown in Fig. 2a

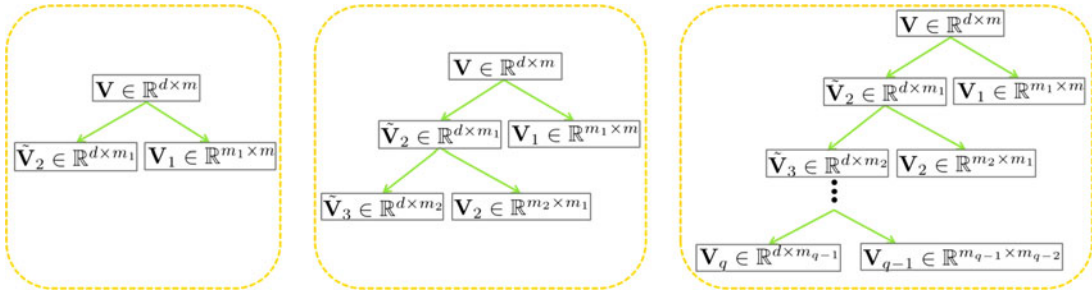
$$\mathbf{V} \approx \tilde{\mathbf{V}}_2 \mathbf{V}_1, \tag{2}$$

where m_1 is the number of latent sub-categories in the 2-nd layer and $\mathbf{V}_1$ indicates the affiliation of m items to m_1 latent sub-categories. $\tilde{\mathbf{V}}_2$ denotes the latent representations of the m_1 sub-categories. Since $\mathbf{V}_1(:,i)$ is the affiliation of the item v_i to the m_1 latent items, then $\tilde{\mathbf{V}}_2\mathbf{V}_1(:,i)$ gives the latent features of v_i. In other words, $\tilde{\mathbf{V}}_2\mathbf{V}_1$ represents the latent features of the m-items.

Since $\tilde{\mathbf{V}}_2$ is non-negative, we can further decompose the representation matrix $\tilde{\mathbf{V}}_2$ to $\mathbf{V}_2 \in \mathbf{R}^{m_2 \times m_1}$ and $\tilde{\mathbf{V}}_3 \in \mathbf{R}^{d \times m_2}$ to get a 3-layer implicit hierarchical structure of items as shown in Fig. 2b

$$\mathbf{V} \approx \tilde{\mathbf{V}}_3 \mathbf{V}_2 \mathbf{V}_1, \tag{3}$$

In particular, $\tilde{\mathbf{V}}_3\mathbf{V}_2\mathbf{V}_1 \in \mathbb{R}^{d \times m}$ denotes the representation of m items, $\tilde{\mathbf{V}}_3\mathbf{V}_2 \in \mathbb{R}^{d \times m_1}$ means the representation of m_1 sub-categories and $\tilde{\mathbf{V}}_3 \in \mathbb{R}^{d \times m_2}$ denotes the representation of m_2 categories. An illustration that the factorization, i.e.,



(a) 2 Layers (b) 3 Layers (c) q Layers

Fig. 2. Implicit hierarchical structures of items via deeply factorizing the item characteristic matrix.

$\tilde{\mathbf{V}}_3\mathbf{V}_2\mathbf{V}_1$, $\tilde{\mathbf{V}}_3\mathbf{V}_2$ and $\tilde{\mathbf{V}}_3$, can describe a 3-layer hierarchical structures is shown in Fig. 4.

Let $\tilde{\mathbf{V}}_{q-1}$ be the latent category affiliation matrix for the $(q-1)$-layer implicit hierarchical structure. The aforementioned process can be generalized to get the q-layer implicit hierarchical structure from $(q-1)$-layer implicit hierarchical structure by further factorizing $\tilde{\mathbf{V}}_{q-1}$ into two non-negative matrices as shown in Fig. 2c

$$\mathbf{V} \approx \mathbf{V}_q\mathbf{V}_{q-1}\dots\mathbf{V}_2\mathbf{V}_1, \tag{4}$$

where $\mathbf{V}_i \in \mathbb{R}^{m_i\times m_{i-1}}$, $(1 < i < q)$ and $\mathbf{V}_q \in \mathbb{R}^{d\times m_{q-1}}$.

Similarly, to model a p-layer user implicit hierarchical structure, we can perform a deep factorization on $\mathbf{U}$ as

$$\mathbf{U} \approx \mathbf{U}_1\mathbf{U}_2\dots\mathbf{U}_{p-1}\mathbf{U}_p, \tag{5}$$

where $\mathbf{U}_1$ is a $n \times n_1$ matrix, $\mathbf{U}_i$ $(1 < i < p)$ is a $n_{i-1} \times n_i$ matrix and $\mathbf{U}_p$ is a $n_{p-1} \times d$ matrix.

3.3 The Proposed IHSR

With model components to model implicit hierarchical structures of items and users, the framework IHSR is proposed to solve the following optimization problem

$$\begin{aligned}\min_{\mathbf{U}_1,\dots,\mathbf{U}_p,\mathbf{V}_1,\dots,\mathbf{V}_q} & ||\mathbf{W}\odot(\mathbf{X}-\mathbf{U}_1\dots\mathbf{U}_p\mathbf{V}_q\dots\mathbf{V}_1)||_F^2 \\ & +\lambda\left(\sum_{i=1}^{p}||\mathbf{U}_i||_F^2+\sum_{j=1}^{q}||\mathbf{V}_j||_F^2\right) \\ s.t.\quad & \mathbf{U}_i \geq \mathbf{0},\ i\in\{1,2,\dots,p\}, \\ & \mathbf{V}_j \geq \mathbf{0},\ j\in\{1,2,\dots,q\},\end{aligned} \tag{6}$$

An illustration of the proposed framework IHSR is demonstrated in Fig. 3. The proposed framework IHSR performs a deep factorizations on the user preference matrix $\mathbf{U}$ and the item characteristic matrix $\mathbf{V}$ to model implicit hierarchical structures of items and users, respectively; while the original WNMF based recommender system only models flat structures as shown in the inner dashed box in Fig. 3.

4 The Proposed Framework for Explicit Hierarchical Structures

When hierarchical structures are explicitly available, it would be superior to incorporate this explicit hierarchical structures than using implicit hierarchical structures because explicit hierarchical structures contain more structural information for guiding the learning process of learning latent features. Therefore, in this section, we introduce how to incorporate explicit hierarchical structures. We aim to provide a unified and flexible framework that can deal with both explicit and implicit hierarchical structures. Thus, we will extend IHSR to incorporate explicit hierarchical structures, which provides us a unified framework. Next, we will first introduce how to mathematically represent explicit hierarchical structures followed by the details of how to incorporate explicit hierarchical structures.

4.1 Representing Explicit Hierarchical Structures

Fig. 4 gives an example of an explicit hierarchical structure. Generally, the hierarchical structure can be regarded as a set of trees, where each node is a group of items that share certain properties. The set of items in a child node is a subset of items in its parent node. Each parent node can have one or more child nodes and each leave node is an item. For example, in Fig. 4, there are two trees, which corresponds to two large categories such as "Kitchen Products" and "Bedroom Products", respectively. The root of the left tree has four items, i.e., $\{v_1, v_2, v_3, v_4\}$, which are all "Kitchen Products". It has two child nodes, which has item sets $\{v_1, v_2\}$ and $\{v_3, v_4\}$, corresponding to two subcategories, i.e., "Cookware" and "Bakeware". From this observation, we can see that a hierarchical structure is simply a set of parent-child relationship, which can be represented as membership matrices. Specifically, let $\mathbf{Q}_k \in \mathbb{R}^{m_{k-1}\times m_k}$ be the membership matrix, which indicate the membership of the child nodes in height k to the nodes in height $k+1$, where the leaves are in height 1. $\mathbf{Q}_k(i,j) = 1$ means that the ith node in height k belong to the j node in height $k+1$. For example, in Fig. 4, we have $\mathbf{Q}_2 \in \mathbb{R}^{m_1\times m_2}$ with $m_1 = 4$ and $m_2 = 2$ as there are 4 nodes in height 2 and 2 nodes in height 3. In addition, $\mathbf{Q}_2(1,1) = 1$, $\mathbf{Q}_2(2,1) = 1$, $\mathbf{Q}_2(3,2) = 1$ and $\mathbf{Q}_2(4,2) = 1$. Note that we assume each tree in the same

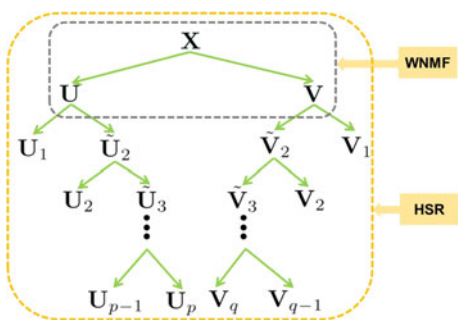


Fig. 3. An illustration of the proposed framework IHSR.

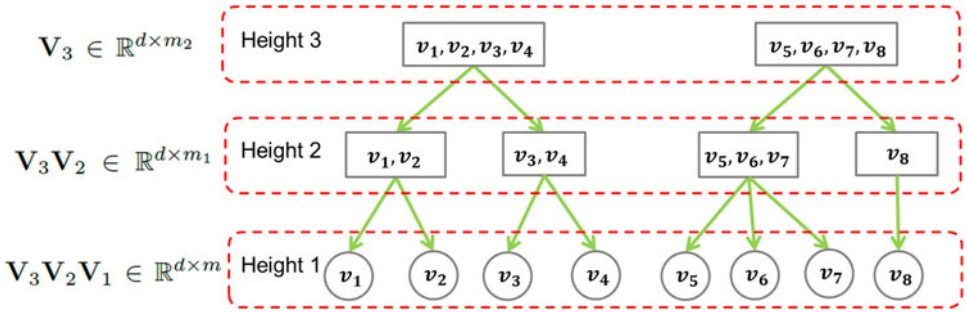


Fig. 4. An illustration of hierarchical structures and feature representation of each nodes.

hierarchical structure has the same depth and the depth from the root node to each leaf node is the same. Otherwise, we can add dummy nodes to realize this. Fig. 5 gives two example of how to add dummy nodes to satisfy the requirement. In Fig. 5a, the left subtree of the root node has depth 2 wile the right subtree has depth 1, thus, we can add one dummy node (i.e., the red node in the figure) to ensure that the left and right subtrees have the same depth. In Fig. 5b, we want a tree of depth 3, thus, we can add a dummy node (red) as shown in the figure. Note that, adding dummy node doesn't change the properties or semantic meanings contained in the hierarchical structure. Instead, adding dummy nodes will make the modeling easier, which will be explained in next section.

Similarly, user preferences also have hierarchical structures. We add dummy nodes to make sure that each tree within the same hierarchical structure has the same depth and the depth from root node to each leaf node is the same. We then use $\mathbf{P}_k \in \mathbb{R}^{n_k \times n_{k-1}}$ to denote the membership matrix for user preference hierarchical structure, which is defined in a similar way. We omit the detail here.

4.2 Modeling Explicit Hierarchical Structures

As observed in Fig. 4, a hierarchical structure can be regarded as a set of trees, where each node of a tree is a group of objects that share certain properties. The items contained in a child node is a subset of the items contained in its parent node. Thus, a parent node captures more general properties for a large set and the child node captures more fine-grained properties for a smaller set. The leaves capture the unique properties of each item. This property implies that the representation of a parent node can be captured by the average of the representations of its child nodes. From the modeling of implicit hierarchical structures, we already have the representation of each node of these trees, i.e., let p be the number of layers of the hierarchical structure, then $\mathbf{V}_p \dots \mathbf{V}_k \in \mathbb{R}^{d\times m_k}$, $k \le p$ represent the m_k nodes in the level $p-k+1$ of a hierarchical structure. For example, as shown in the Fig. 4, $\mathbf{V}_3\mathbf{V}_2\mathbf{V}_1 \in \mathbb{R}^{d\times m}$ with $m = 8$ denotes the latent features of the eight leave nodes; $\mathbf{V}_3\mathbf{V}_2 \in \mathbb{R}^{d\times m_1}$ with $m_1 = 4$ denotes the latent representations of the four nodes; and $\mathbf{V}_3 \in \mathbb{R}^{d\times m_2}$ gives the latent representations of the two root nodes. Since the representation of a parent node can be captured by the average of the representations of its child nodes, and the parent child relationship is captured in $\mathbf{Q}_k$, it is easy to model this relationship. Specifically, we first normalize $\mathbf{Q}_k$ as

$$\mathbf{Q}_k(i,j) \leftarrow \frac{\mathbf{Q}_k(i,j)}{\sum_{i=1}^{m_{k-1}} \mathbf{Q}_k(i,j)}. \tag{7}$$

Then the requirement that the representation of a parent node should be close to the average of the representations of its child nodes can be mathematically modeled as

$$\min \sum_{i=2}^{q} \|\mathbf{V}_q \dots \mathbf{V}_i - \mathbf{V}_q \dots \mathbf{V}_{i-1}\mathbf{Q}_{i-1}\|_F^2. \tag{8}$$

Similarly, we do the normalization on $\mathbf{P}_k$ as

$$\mathbf{P}_k(i,j) \leftarrow \frac{\mathbf{P}_k(i,j)}{\sum_{j=1}^{n_{k-1}} \mathbf{P}_k(i,j)}. \tag{9}$$

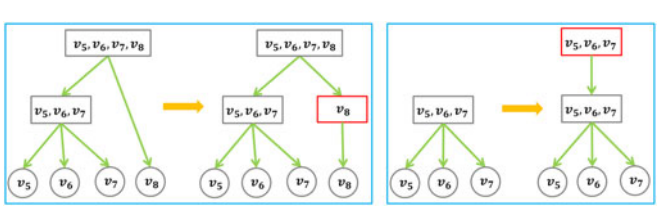
With the normalized $\mathbf{P}_k$, we model the explicit hierarchical structure for user preferences as

$$\min \sum_{i=2}^{p} \|\mathbf{U}_i \dots \mathbf{U}_p - \mathbf{P}_{i-1}\mathbf{U}_{i-1} \dots \mathbf{U}_p\|_F^2. \tag{10}$$

4.3 The Proposed HSR

With the model components to model explicit hierarchical structures of items and users, the framework HSR is proposed to solve the following optimization problem

$$\begin{aligned} \min_{\mathbf{U}_1,\dots,\mathbf{U}_p,\mathbf{V}_1,\dots,\mathbf{V}_q} & \|\mathbf{W} \odot (\mathbf{X} - \mathbf{U}_1 \dots \mathbf{U}_p\mathbf{V}_q \dots \mathbf{V}_1)\|_F^2 \\ & + \alpha \sum_{i=2}^{p} \|\mathbf{U}_i \dots \mathbf{U}_p - \mathbf{P}_{i-1}\mathbf{U}_{i-1} \dots \mathbf{U}_p\|_F^2 \\ & + \beta \sum_{i=2}^{q} \|\mathbf{V}_q \dots \mathbf{V}_i - \mathbf{V}_q \dots \mathbf{V}_{i-1}\mathbf{Q}_{i-1}\|_F^2 \\ & + \lambda \left(\sum_{i=1}^{p} \|\mathbf{U}_i\|_F^2 + \sum_{j=1}^{q} \|\mathbf{V}_j\|_F^2 \right) \\ s.t. \quad & \mathbf{U}_i \ge \mathbf{0},\ i \in \{1,2,\dots,p\}, \\ & \mathbf{V}_j \ge \mathbf{0},\ j \in \{1,2,\dots,q\}, \end{aligned} \tag{11}$$



(a) Dummy Node Example 1 (b) Dummy Node Example 2

Fig. 5. Adding dummy nodes. In each example, the left tree is the original tree and the right tree is the tree with dummy node. The red node is dummy node. (Best viewed in color).

where α and β are two parameters to control the contribution of the explicit user and item hierarchical structures, respectively. The proposed framework is a general framework that can deal with both explicit and implicit hierarchical structures. When explicit hierarchical structures are available, α and β are set to positive values to guide the learning process of user and item latent features. When explicit hierarchical structures are not available, we can set α and β to zero, then HSR reduces to IHSR, which models implicit hierarchical structures. For mixed hierarchical structures such as explicit item HS and implicit user HS, we can set α to be 0 and β to be positive.

4.4 Discussion of Dummy Nodes

For simplicity, we will use the example in Fig. 5a to show that dummy nodes don't affect the modeling purpose. Our modeling follows the assumption that a parent node's representation is close to the average of its child nodes' representations. Thus, we only need to show that with and without the dummy nodes, we achieve the same goal. Consider the left tree without dummy nodes in Fig. 5a, we would require the representation of node $\{v_5, v_6, v_7, v_7\}$ to be close to the average representations of nodes $\{v_5, v_6, v_7\}$ and node $\{v_8\}$. After adding the dummy nodes, i.e., the right tree in Fig. 5a, we would require the representation of node $\{v_5, v_6, v_7, v_7\}$ to be close to the average representation of node $\{v_5, v_6, v_7\}$ and node $\{v_8\}$ (red circle). In addition, we also have that the representation of node $\{v_8\}$ (red rectangle) is close to that of $\{v_8\}$ (gray circle). The total effect is that we are enforcing the representation of node $\{v_5, v_6, v_7, v_7\}$ to be close to the average representation of node $\{v_5, v_6, v_7\}$ and node $\{v_8\}$ (gray circle). Thus, the modeling effects with and without dummy nodes are approximately the same. However, by adding dummy nodes, we make the modeling easier because now each tree has similar form and the number of leaves of all the trees are equal to number of items. The modeling of hierarchical structures can be written in the simple form as Eq. (11).

5 An Optimization Framework

The objective function in Eq. (11) is not convex if we update all the variable jointly but it is convex if we update the variables alternatively. We will first introduce our optimization method for HSR based on an alternating scheme in [40] and then we will give convergence analysis and complexity analysis of the optimization method.

5.1 Inferring Parameters of HSR

In this section, we give the details of updating rules of $\mathbf{U}_i$ and $\mathbf{V}_i$.

5.1.1 Update Rule of $\mathbf{U}_i$

To update $\mathbf{U}_i$, we fix the other variables except $\mathbf{U}_i$. By removing terms that are irrelevant to $\mathbf{U}_i$, Eq. (11) can be rewritten as

$$\begin{aligned}\min_{\mathbf{U}_i \geq \mathbf{0}} &\|\mathbf{W} \odot (\mathbf{X} - \mathbf{A}_i\mathbf{U}_i\mathbf{H}_i)\|_F^2 + \alpha\|(\mathbf{I} - \mathbf{P}_i\mathbf{U}_i)\mathbf{D}_i\|_F^2 \\ &+ \alpha \sum_{k=1}^{i-1} \|(\mathbf{G}_i^k - \mathbf{C}_i^k)\mathbf{U}_i\mathbf{D}_i\|_F^2 + \lambda\|\mathbf{U}_i\|_F^2\end{aligned} \tag{12}$$

where $\mathbf{A}_i$, $\mathbf{H}_i$, and $\mathbf{D}_i$, $1 \leq i \leq p$, are defined as

$$\mathbf{A}_i = \begin{cases} \mathbf{U}_1\mathbf{U}_2 \dots \mathbf{U}_{i-1} & \text{if } i \neq 1 \\ \mathbf{I} & \text{if } i = 1 \end{cases} \tag{13}$$

$$\mathbf{H}_i = \begin{cases} \mathbf{U}_{i+1} \dots \mathbf{U}_p\mathbf{V}_q \dots \mathbf{V}_1 & \text{if } i \neq p \\ \mathbf{V}_q \dots \mathbf{V}_1 & \text{if } i = p \end{cases} \tag{14}$$

$$\mathbf{D}_i = \begin{cases} \mathbf{U}_{i+1}\mathbf{U}_{i+2} \dots \mathbf{U}_p & \text{if } i \neq p \\ \mathbf{I} & \text{if } i = p \end{cases} \tag{15}$$

$\mathbf{G}_i^k$ and $\mathbf{C}_i^k$, $1 \leq i \leq p$, $1 \leq k \leq i-1$, are defined as

$$\mathbf{G}_i^k = \begin{cases} \mathbf{U}_{k+1} \dots \mathbf{U}_{i-1} & \text{if } k \neq i-1 \\ \mathbf{I} & \text{if } k = i-1 \end{cases} \tag{16}$$

$$\mathbf{C}_i^k = \mathbf{P}_k\mathbf{U}_k\mathbf{G}_i^k \tag{17}$$

The Lagrangian function of Eq. (12) is

$$\begin{aligned}\mathcal{L}(\mathbf{U}_i) = &\|\mathbf{W} \odot (\mathbf{X} - \mathbf{A}_i\mathbf{U}_i\mathbf{H}_i)\|_F^2 + \lambda\|\mathbf{U}_i\|_F^2 - Tr(\mathbf{Y}^T\mathbf{U}_i) \\ &+ \alpha\|(\mathbf{I} - \mathbf{P}_i\mathbf{U}_i)\mathbf{D}_i\|_F^2 + \alpha \sum_{k=1}^{i-1} \|(\mathbf{G}_i^k - \mathbf{C}_i^k)\mathbf{U}_i\mathbf{D}_i\|_F^2,\end{aligned} \tag{18}$$

where $\mathbf{Y}$ is the Lagrangian multiplier. The derivative of $\mathcal{L}(\mathbf{U}_i)$ with respect to $\mathbf{U}_i$ is

$$\begin{aligned}\frac{\partial \mathcal{L}(\mathbf{U}_i)}{\partial \mathbf{U}_i} = &2\mathbf{A}_i^T[\mathbf{W} \odot (\mathbf{A}_i\mathbf{U}_i\mathbf{H}_i - \mathbf{X})]\mathbf{H}_i^T + 2\lambda\mathbf{U}_i \\ &+ 2\alpha\mathbf{P}_i^T(\mathbf{P}_i\mathbf{U}_i\mathbf{D}_i - \mathbf{D}_i)\mathbf{D}_i^T - \mathbf{Y} \\ &+ 2\alpha \sum_{k=1}^{i-1} (\mathbf{G}_i^k - \mathbf{C}_i^k)^T(\mathbf{G}_i^k - \mathbf{C}_i^k)\mathbf{U}_i\mathbf{D}_i\mathbf{D}_i^T\end{aligned} \tag{19}$$

By setting the derivative to zero and using Karush-Kuhn-Tucker complementary condition [41], i.e., $\mathbf{Y}(s,t)\mathbf{U}_i(s,t) = 0$, we get

$$\begin{aligned}\Big[&\mathbf{A}_i^T[\mathbf{W} \odot (\mathbf{A}_i\mathbf{U}_i\mathbf{H}_i - \mathbf{X})]\mathbf{H}_i^T + \alpha \sum_{k=1}^{i-1} (\mathbf{G}_i^k - \mathbf{C}_i^k)^T(\mathbf{G}_i^k - \mathbf{C}_i^k) \\ &\mathbf{U}_i\mathbf{D}_i\mathbf{D}_i^T + \alpha\mathbf{P}_i^T(\mathbf{P}_i\mathbf{U}_i\mathbf{D}_i - \mathbf{D}_i)\mathbf{D}_i^T + \lambda\mathbf{U}_i\Big](s,t)\mathbf{U}_i(s,t) = 0\end{aligned} \tag{20}$$

Eq.(20) leads to the following update rule of $\mathbf{U}_i$ as

$$\mathbf{U}_i(s,t) \leftarrow \mathbf{U}_i(s,t)\sqrt{\frac{\mathbf{E}_i^1(s,t)}{\mathbf{E}_i^2(s,t)}} \tag{21}$$

where $\mathbf{E}_i^1$ and $\mathbf{E}_i^2$ are

$$\begin{aligned}\mathbf{E}_i^1 = &\mathbf{A}_i^T(\mathbf{W} \odot \mathbf{X})\mathbf{H}_i^T + \alpha\mathbf{P}_i^T\mathbf{D}_i\mathbf{D}_i^T \\ &+ \alpha \sum_{k=1}^{i-1} (\mathbf{G}_i^k\mathbf{C}_i^{kT} + \mathbf{C}_i^{kT}\mathbf{G}_i^k)\mathbf{U}_i\mathbf{D}_i\mathbf{D}_i^T \\ \mathbf{E}_i^2 = &\mathbf{A}_i^T(\mathbf{W} \odot (\mathbf{A}_i\mathbf{U}_i\mathbf{H}_i))\mathbf{H}_i^T + \alpha\mathbf{P}_i^T\mathbf{P}_i\mathbf{U}_i\mathbf{D}_i\mathbf{D}_i^T \\ &+ \alpha \sum_{k=1}^{i-1} (\mathbf{G}_i^{kT}\mathbf{G}_i^k + \mathbf{C}_i^{kT}\mathbf{C}_i^k)\mathbf{U}_i\mathbf{D}_i\mathbf{D}_i^T + \lambda\mathbf{U}_i\end{aligned} \tag{22}$$

5.1.2 Update Rule of $\mathbf{V}_i$

Similarly, to update $\mathbf{V}_i$, we fix the other variables except $\mathbf{V}_i$. By removing terms that are irrelevant to $\mathbf{V}_i$, the optimization problem for $\mathbf{V}_i$ is

$$\begin{aligned}\min_{\mathbf{V}_i\geq\mathbf{0}}&\|\mathbf{W}\odot(\mathbf{X}-\mathbf{B}_i\mathbf{V}_i\mathbf{M}_i)\|_F^2+\beta\|\mathbf{T}_i(\mathbf{I}-\mathbf{V}_i\mathbf{Q}_i)\|_F^2\\&+\beta\sum_{k=1}^{i-1}\|\mathbf{T}_i\mathbf{V}_i(\mathbf{L}_i^k-\mathbf{S}_i^k)\|_F^2+\lambda\|\mathbf{V}_i\|_F^2,\end{aligned}\tag{23}$$

where $\mathbf{B}_i$, $\mathbf{M}_i$ and $\mathbf{T}_i$, $1\leq i\leq q$, are defined as

$$\mathbf{B}_i=\begin{cases}\mathbf{U}_1\dots\mathbf{U}_p\mathbf{V}_q\dots\mathbf{V}_{i+1} & \text{if } i\neq q\\ \mathbf{U}_1\dots\mathbf{U}_p & \text{if } i=q\end{cases}\tag{24}$$

$$\mathbf{M}_i=\begin{cases}\mathbf{V}_{i-1}\dots\mathbf{V}_1 & \text{if } i\neq 1\\ \mathbf{I} & \text{if } i=1\end{cases}\tag{25}$$

$$\mathbf{T}_i=\begin{cases}\mathbf{V}_q\dots\mathbf{V}_{i+1} & \text{if } i\neq 1\\ \mathbf{I} & \text{if } i=1\end{cases}\tag{26}$$

$\mathbf{L}_i^k$ and $\mathbf{S}_i^k$, $1\leq i\leq q$, $1\leq k\leq i-1$, are defined as

$$\mathbf{L}_i^k=\begin{cases}\mathbf{V}_{i-1}\dots\mathbf{V}_{k+1} & \text{if } k\neq i-1\\ \mathbf{I} & \text{if } k=i-1\end{cases}\tag{27}$$

$$\mathbf{S}_i^k=\mathbf{L}_i^k\mathbf{V}_k\mathbf{Q}_k\tag{28}$$

The Lagrangian function of Eq. (23) is

$$\begin{aligned}\min_{\mathbf{V}_i\geq\mathbf{0}}&\|\mathbf{W}\odot(\mathbf{X}-\mathbf{B}_i\mathbf{V}_i\mathbf{M}_i)\|_F^2+\lambda\|\mathbf{V}_i\|_F^2-Tr(\mathbf{Z^T}\mathbf{V_i})\\&+\beta\|\mathbf{T}_i(\mathbf{I}-\mathbf{V}_i\mathbf{Q}_i)\|_F^2+\beta\sum_{k=1}^{i-1}\|\mathbf{T}_i\mathbf{V}_i(\mathbf{L}_i^k-\mathbf{S}_i^k)\|_F^2,\end{aligned}\tag{29}$$

where $\mathbf{Z}$ is the Lagrangian multiplier. The derivative of $\mathcal{L}(\mathbf{V}_i)$ with respect to $\mathbf{V}_i$ is

$$\begin{aligned}\frac{\partial\mathcal{L}(\mathbf{V}_i)}{\partial\mathbf{V}_i}&=2\mathbf{B}_i^T[\mathbf{W}\odot(\mathbf{B}_i\mathbf{V}_i\mathbf{M}_i-\mathbf{X})]\mathbf{M}_i^T+2\lambda\mathbf{V}_i\\&+2\beta\mathbf{T}_i^T(\mathbf{T}_i\mathbf{V}_i\mathbf{Q}_i-\mathbf{T}_i)\mathbf{Q}_i^T-\mathbf{Z}\\&+2\beta\sum_{k=1}^{i-1}\mathbf{T}_i^T\mathbf{T}_i\mathbf{V}_i(\mathbf{L}_i^k-\mathbf{S}_i^k)(\mathbf{L}_i^k-\mathbf{S}_i^k)^T.\end{aligned}\tag{30}$$

Similarly, by setting the derivative to zero and using Karush-Kuhn-Tucker complementary condition, $\mathbf{Z}(s,t)\mathbf{V}_i(s,t)=0$, we arrive at

$$\begin{aligned}&\Big[\mathbf{B}_i^T[\mathbf{W}\odot(\mathbf{B}_i\mathbf{V}_i\mathbf{M}_i-\mathbf{X})]\mathbf{M}_i^T+\beta\sum_{k=1}^{i-1}\mathbf{T}_i^T\mathbf{T}_i\mathbf{V}_i(\mathbf{L}_i^k-\mathbf{S}_i^k)\\&(\mathbf{L}_i^k-\mathbf{S}_i^k)^T+\beta\mathbf{T}_i^T(\mathbf{T}_i\mathbf{V}_i\mathbf{Q}_i-\mathbf{T}_i)\mathbf{Q}_i^T+\lambda\mathbf{V}_i\Big](s,t)\mathbf{V}_i(s,t)=0,\end{aligned}\tag{31}$$

which leads to the following update rule of $\mathbf{V}_i$ as

$$\mathbf{V}_i(s,t)\leftarrow\mathbf{V}(s,t)\sqrt{\frac{\mathbf{F}_i^1(s,t)}{\mathbf{F}_i^2(s,t)}}\tag{32}$$

where $\mathbf{F}_i^1$ and $\mathbf{F}_i^2$ are given as

$$\begin{aligned}\mathbf{F}_i^1&=\mathbf{B}_i^T(\mathbf{W}\odot\mathbf{X})\mathbf{M}_i^T+\beta\mathbf{T}_i^T\mathbf{T}_i\mathbf{Q}_i^T\\&+\beta\sum_{k=1}^{i-1}\mathbf{T}_i^T\mathbf{T}_i\mathbf{V}_i(\mathbf{L}_i^k\mathbf{S}_i^{kT}+\mathbf{L}_i^{kT}\mathbf{S}_i^k)\end{aligned}\tag{33}$$

$$\begin{aligned}\mathbf{F}_i^2&=\mathbf{B}_i^T[\mathbf{W}\odot(\mathbf{B}_i\mathbf{V}_i\mathbf{M}_i)]\mathbf{M}_i^T+\beta\mathbf{T}_i^T\mathbf{T}_i\mathbf{V}_i\mathbf{Q}_i\mathbf{Q}_i^T\\&+\beta\sum_{k=1}^{i-1}\mathbf{T}_i^T\mathbf{T}_i\mathbf{V}_i(\mathbf{L}_i^k(\mathbf{L}_i^k)^T+\mathbf{S}_i^k(\mathbf{S}_i^k)^T)+\lambda\mathbf{V}_i.\end{aligned}\tag{34}$$

5.2 Learning Algorithm for HSR

With the update rules for $\mathbf{U}_i$ and $\mathbf{V}_j$, the optimization algorithm for HSR is shown in Algorithm 1. Next we briefly review Algorithm 1. In order to expedite the approximation of the factors in HSR, we pre-train each layer to have an initial approximation of the matrices $\mathbf{U}_i$ and $\mathbf{V}_i$. To perform pre-training, we first use WNMF [3] to decompose the user-item rating matrix into $\tilde{\mathbf{U}}_1\tilde{\mathbf{V}}_1$ by solving Eq. (1). After that, we further decompose $\tilde{\mathbf{U}}_1$ into $\tilde{\mathbf{U}}_1\approx\mathbf{U}_1\tilde{\mathbf{U}}_2$ and $\tilde{\mathbf{V}}_1\approx\tilde{\mathbf{V}}_2\mathbf{V}_1$ using nonnegative matrix factorization. We keep the decomposition process until we have p user layers and q item layers. This initializing process is summarized in Algorithm 1 from line 1 to line 9. After initialization, we will do fine-tuning by updating the $\mathbf{U}_i$ and $\mathbf{V}_i$ using updating rules in Eq. (21) and Eq. (32) separately. The procedure is to first update $\mathbf{V}_i$ in sequence and then $\mathbf{U}_i$ in sequence alternatively, which is summarized in Algorithm 1 from line 10 to line 20. In line 21, we reconstruct the user-item matrix as $\mathbf{X}_{pred}=\mathbf{U}_1\dots\mathbf{U}_p\mathbf{V}_q\dots\mathbf{V}_1$. A missing rating from u_i to v_j will be predicted as $\mathbf{X}_{pred}(i,j)$

Algorithm 1. The Optimization Algorithm for the Proposed Framework HSR

Input: $\mathbf{X},\alpha,\beta,\lambda,p,q,d$ and dimensions of each layer
Output: $\mathbf{X}_{pred}$

1: Initialize $\{\mathbf{U}_i\}_{i=1}^p$ and $\{\mathbf{V}_i\}_{i=1}^q$
2: $\tilde{\mathbf{U}}_1,\tilde{\mathbf{V}}_1\leftarrow\mathrm{WNMF}(\mathbf{X},d)$
3: **for** i = 1 to p-1 **do**
4: $\quad\mathbf{U}_i,\tilde{\mathbf{U}}_{i+1}\leftarrow\mathrm{NMF}(\tilde{\mathbf{U}}_i,n_i)$
5: **end for**
6: **for** i = 1 to q-1 **do**
7: $\quad\tilde{\mathbf{V}}_{i+1},\mathbf{V}_i\leftarrow\mathrm{NMF}(\tilde{\mathbf{V}}_i,m_i)$
8: **end for**
9: $\mathbf{U}_p=\tilde{\mathbf{U}}_p$, $\mathbf{V}_q=\tilde{\mathbf{V}}_q$
10: **repeat**
11: $\quad$**for** i = 1 to q **do**
12: $\quad\quad$update $\mathbf{B}_i$, $\mathbf{M}_i$, $\mathbf{T}_i$, $\mathbf{L}_i^k$ and $\mathbf{S}_i^k$, $1\leq k\leq i-1$, with Eqs. (24), (25), (26), (27), and (28)
13: $\quad\quad$update $\mathbf{V}_i$ by Eq. (32)
14: $\quad$**end for**
15:
16: $\quad$**for** i = p to 1 **do**
17: $\quad\quad$update $\mathbf{A}_i$, $\mathbf{H}_i$, $\mathbf{D}_i$, $\mathbf{G}_i^k$, $1\leq k\leq i-1$, and $\mathbf{C}_i^k$ with Eq. (13), (14), (15), (16), and (17)
18: $\quad\quad$update $\mathbf{U}_i$ by Eq. (21)
19: $\quad$**end for**
20: **until** Stopping criterion is reached
21: predict rating matrix $\mathbf{X}_{pred}=\mathbf{U}_1\dots\mathbf{U}_p\mathbf{V}_q\dots\mathbf{V}_1$

5.3 Learning Algorithm for IHSR

The proposed model HSR is a unified and flexible framework which can be used for both implicit and explicit hierarchical

structures. By setting α and β to zero, HSR reduces to IHSR. Thus, the learning algorithm of HSR can also be used for IHSR. Specifically, by setting α and β to zero, Algorithm 1 becomes a learning algorithm for IHSR. We omit the details here.

5.4 Time Complexity Analysis

Initialization and fine-tuning are two most expensive operations for Algorithm 1. For line 3 to 5, the time complexity of factorization of $\tilde{\mathbf{U}}_i \in \mathbb{R}^{n_{i-1} \times d}$ to $\mathbf{U}_i \in \mathbb{R}^{n_{i-1} \times n_i}$ and $\tilde{\mathbf{U}}_{i+1} \in \mathbb{R}^{n_i \times d}$ is $\mathcal{O}(tn_{i-1}n_i d)$ for $1 < i < p$, and $\mathcal{O}(tnn_1 d)$ for $i = 1$, where t is number of iterations takes for the decomposition. Thus the cost of initializing $\mathbf{U}_i$'s is $\mathcal{O}(td(nn_1 + n_1n_2 + \ldots + n_{p-2}n_{p-1})$. Similarly, the cost of initializing $\mathbf{V}_i$'s is $\mathcal{O}(td(mm_1 + m_1m_2 + \ldots + m_{q-2}m_{q-1})$ (line 6 to 8). The computational cost of fine-tuning $\mathbf{U}_i$ in each iteration is $\mathcal{O}\big(nn_{i-1}n_i + nn_im + n_{i-1}n_im + n_i^2(d + n_{i-1}) + n_{i-1}^2(n_i \sum_{k=1}^{i-1} n_k)\big)$. Similarly, the computational cost of fine-tuning $\mathbf{V}_i$ in each iteration is $\mathcal{O}\big(mm_{i-1}m_i + mm_in + m_{i-1}m_in + m_i^2(d + m_{i-1}) + m_{i-1}^2(m_i + \sum_{k=1}^{i-1} m_k)\big)$. Let $n_0 = n, m_0 = m, n_p = m_q = d$, then the time complexity of fine-tuning for HSR is $\mathcal{O}\big(t_f[(n+m)(\sum_{i=1}^{p} n_{i-1}n_i + \sum_{j=1}^{q} m_{i-1}m_i) + nm(\sum_{i=1}^{p} n_i + \sum_{j=1}^{q} m_j) + \sum_{i=1}^{p} n_i^2(d + n_{i-1}) + \sum_{i=1}^{p} n_{i-1}^2(n_i \sum_{k=1}^{i-1} n_k) + \sum_{i=1}^{q} m_i^2(d + m_{i-1}) + \sum_{i=1}^{q} m_{i-1}^2(m_i + \sum_{k=1}^{i-1} m_k)]\big)$, where t_f is the number of iterations takes to fine-tune. For IHSR, since we set $\alpha = 0$ and $\beta = 0$, the time complexity of fine-tuning for IHSR is $\mathcal{O}\big(t_f[(n+m)(\sum_{i=1}^{p} n_{i-1}n_i + \sum_{j=1}^{q} m_{i-1}m_i) + nm(\sum_{i=1}^{p} n_i + \sum_{j=1}^{q} m_j)]\big)$. The overall time complexity is the summation of the costs of initialization and fine-tuning. Note that in practice, for both IHSR and HSR, two layers of users and items, i.e., $p = 2$ and $q = 2$, already give significant performance improvement over MF and NMF. When p and q are both set to a larger value than 2, the performance is slightly better than that of $p = 2$ and $q = 2$ but the time complexity increases. *Thus, in practice, we usually choose $p = 2$ and $q = 2$ and thus the time complexity is not that large compared with WNMF and MF, while the performance improvement is significant enough.*

5.5 Convergence Analysis

The convergence of the algorithm is guaranteed. The details of the convergence analysis can be found in the supplementary material available from the first author's homepage.

6 Experimental Analysis

In this section, we conduct experiments to evaluate the effectiveness of the proposed framework IHSR/HSR and factors that could affect the performance. We begin by introducing datasets and experimental settings, then we compare the proposed framework with the state-of-the-art recommendation systems. Further experiments are conducted to investigate the parameter sensitivity of IHSR and HSR.

6.1 Datasets

The experiments are conducted on four publicly available benchmark datasets, i.e., MovieLens100K,[3] Douban,[4] MovieLens1m[5] and Ciao.[6]

TABLE 1
Statistics of the Datasets

Dataset	# of users	# of items	# of ratings
MovieLens100K	943	1,682	100,000
Douban	1,371	1,967	149,623
MovieLens1M	6,040	3,900	1,000,209
Ciao	7,935	16,200	171,465

- MovieLens100K: MovieLens100K consists of 100,000 movie ratings of 943 users for 1,682 movies. The movies are categorized into 18 categories, such as "Animation", "Children's" and "Comedy".
- Douban: The Douban dataset is crawled from Douban[7] and used in [42]. We filter out users who rated less than 20 movies and movies that are rated by less than 10 users from the Douban dataset and get a dataset consisting of 149,623 movie ratings of 1,371 users and 1,967 movies.
- MovieLens1M: MovieLens1M is a widely used movie rating datasets, which contains around 1 million anonymous ratings of approximately 3,900 movies by 6,040 users. Each user rated at least 20 items. Similarly, the movies are also categorized into 18 categories.
- Ciao: The Ciao dataset is collected from a real-world social media websites, Ciao.[8] From the originally collected dataset, we filter out users who rated few items and also items that received less than 3 ratings, which leaves us 7,935 users and 16,200 products. Each product belongs to one of the 30 categories defined by Ciao, such as "Electronics", "Home & Garden" and "Fashion". Each category is further divided into several sub-categories.

For all four datasets, users can rate items with scores from 1 to 5. For MovieLens100K, MovieLens1M and Ciao, in addition to rating matrix, we also have the category information of the items. The category information are used as the explicit hierarchical structures for the evaluation of HSR. The statistics of datasets are summarized in Table 1. We adopt these datasets because: (i) they are widely used for evaluating the effectiveness of recommender systems; and (ii) three of these datasets have category information, which can be used for the evaluation of HSR.

6.2 Evaluation Settings

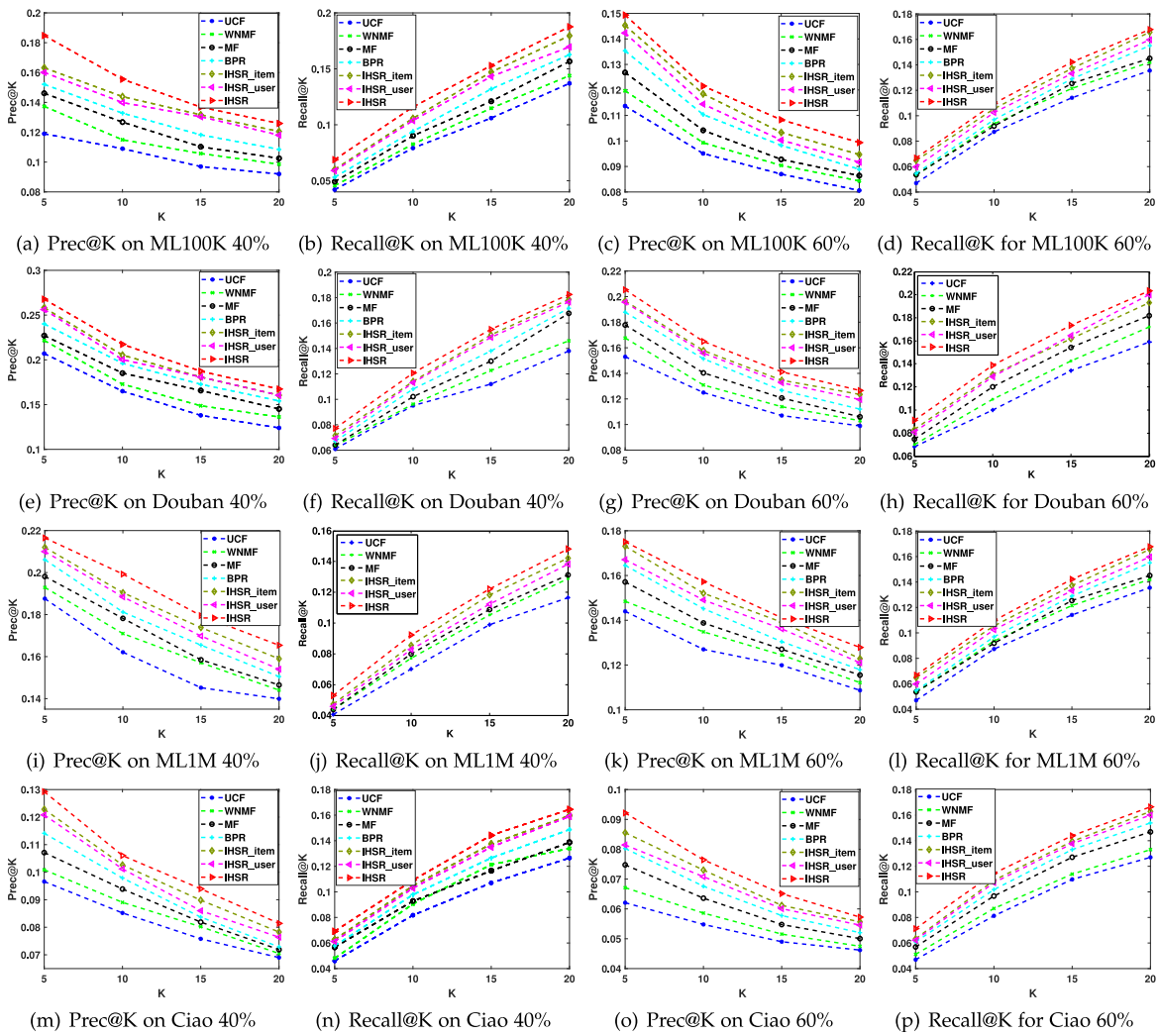
We perform both rating prediction and Top-K recommendation to evaluate the effectiveness of the proposed framework. For rating prediction, the widely used evaluation metric, i.e., root mean square error (RMSE), is adopted to evaluate the performance. For Top-K recommendation, we adopt precision@K (prec@K) and recall@K as the evaluation metrics, where K is varied as $\{5, 10, 15, 20\}$.

We random select x percent as training set and the remaining $1 - x$ percent as test set where x is varied as $\{40, 60\}$. *Note that for rating prediction, the test set contains ratings from 1 to 5; while for top-K ranking, following previous work [27], we remove ratings below 5-stars from test set such that the test set contains only 5-stars ratings. Thus, we can reasonably state that test set*

3. http://grouplens.org/datasets/movielens/100k/
4. http://dl.dropbox.com/u/17517913/Douban.zip
5. https://grouplens.org/datasets/movielens/1m/
6. http://www.cse.msu.edu/ tangjili/trust.html
7. https://www.douban.com/
8. http://www.ciao.co.uk/

TABLE 2
Rating Prediction Results (RMSE±std) on MovieLens100K, Douban, Movielens1M, and Ciao

Methods		UCF	MF	WNMF	IHSR-User	IHSR-Item	IHSR
MovieLens100K	40%	1.0615±0.0032	0.9792±0.0029	1.0205±0.0034	0.9681±0.0028	0.9672±0.0027	0.9578±0.0024
	60%	1.0446±0.0030	0.9664±0.0022	0.9953±0.0025	0.9433±0.0015	0.9412±0.0020	0.9325±0.0032
Douban	40%	0.8077±0.0019	0.7538±0.0023	0.7807±0.0021	0.7313±0.0013	0.7304±0.0016	0.7284±0.0014
	60%	0.7988±0.0028	0.7403±0.0025	0.7637±0.0023	0.7225±0.0014	0.7219±0.0017	0.7179±0.0018
MovieLens1M	40%	1.0842±0.0007	0.9306±0.0004	0.9519±0.0007	0.9021±0.0009	0.9015±0.0008	0.8957±0.0009
	60%	1.0549± 0.0010	0.9085±0.0020	0.9384±0.0012	0.8877±0.0009	0.8882±0.0010	0.8813±0.0009
Ciao	40%	1.1280±0.0031	1.0946±0.0028	1.1188±0.0038	1.0783±0.0021	1.0778±0.0023	1.0695±0.0021
	60%	1.1040±0.0025	1.0702±0.0038	1.0811±0.0024	1.0251±0.0043	1.0260±0.0039	1.0178±0.0032



(a) Prec@K on ML100K 40% (b) Recall@K on ML100K 40% (c) Prec@K on ML100K 60% (d) Recall@K for ML100K 60%

(e) Prec@K on Douban 40% (f) Recall@K on Douban 40% (g) Prec@K on Douban 60% (h) Recall@K for Douban 60%

(i) Prec@K on ML1M 40% (j) Recall@K on ML1M 40% (k) Prec@K on ML1M 60% (l) Recall@K for ML1M 60%

(m) Prec@K on Ciao 40% (n) Recall@K on Ciao 40% (o) Prec@K on Ciao 60% (p) Recall@K for Ciao 60%

Fig. 6. Top K ranking on MovieLens100K, Douban, MovieLens1M, and Ciao.

contains items relevant to the respective users. For both rating prediction and top-K recommendation, the training set contains ratings from 1 to 5. The random selection is carried out 10 times independently, and the average results are reported. A smaller RMSE value means better rating prediction performance; while a larger Prec@K and Recall@K suggests better top-K recommendation performance.

6.3 Recommendation Performance of IHSR

In this section, we conduct experiments to show the effectiveness of exploring implicit hierarchical structures by comparing the recommendation performance of IHSR with representation recommender systems. Note that for IHSR, no explicit hierarchical structures are used. The comparison results for rating prediction and top-K recommendation are summarized in Table 2 and Fig. 6, respectively. The baseline methods in the table and figure are defined as:

- UCF: UCF is the user-oriented collaborative filtering where the rating from u_i to v_j is predicted as an aggregation of ratings of K most similar users of u_i

TABLE 3
Rating Prediction Results (RMSE±std) on MovieLens100K, Movielens1M, and Ciao

Methods		ReMF	IHSR	HSR-Item	HSR
MovieLens100K	40%	0.9751±0.0026	0.9578±0.0024	0.9659±0.0027	0.9558±0.0022
	60%	0.9499±0.0028	0.9325±0.0032	0.9426±0.0033	0.9296±0.0027
MovieLens1M	40%	0.9152±0.0014	0.8957±0.0009	0.8950±0.0006	0.8912±0.0010
	60%	0.8923±0.0010	0.8813±0.0009	0.8798±0.0021	0.8777±0.0019
Ciao	40%	1.0791±0.0021	1.0695±0.0021	1.0687±0.0023	1.0634± 0.0020
	60%	1.0320±0.0034	1.0178±0.0032	1.0166±0.0032	1.0134±0.0027

to v_j. The cosine similarity is used to calculate user-user similarity.

- MF: matrix factorization based collaborative filtering tries to decompose the user-item rating matrix into two matrices such that the reconstruction error is minimized [9]. It is one of the most popular recommender systems.
- WNMF: weighted nonnegative matrix factorization tries to decompose the weighted rating matrix into two nonnegative matrices to minimize the reconstruction error [3]. It is the basic model of the proposed framework.
- BPR: Bayesian personalized ranking [25] is state-of-the-art ranking based method proposed for top-K recommendation. Since it is not good at predicting the ratings, we only use is as a baseline for top-K recommendation.
- IHSR-Item: IHSR-Item is a variant of the proposed framework IHSR. IHSR-Item only considers the implicit hierarchical structure of items by setting $p = 1$.
- IHSR-User: IHSR-User is a variant of the proposed framework IHSR. IHSR-Users only considers the implicit hierarchical structure of users by setting $q = 1$.

The compred baselines include representative and state-of-the-art rating based and ranking based methods. IHSR-Item and IHSR-Users are aim to show that by considering only item hierarchical structures or item hierarchical structures, can we gain improvement. Note that parameters of all methods are determined via cross validation. Based on the results, we make the following observations:

- In general, matrix factorization based recommender systems outperform the user-oriented CF method and this observation is consistent with that in [9].
- Both IHSR-Item and IHSR-User obtain better results than WNMF. We perform t-test on these results, which suggest that the improvement is significant. These results indicate that the implicit hierarchical structures of users and items can improve the recommendation performance.
- BPR outperforms MF on top-K recommendation becuase it is optimized for ranking. IHSR outperforms BPR, which suggests the effectiveness considering hierarchy.
- IHSR consistently outperforms both IHSR-Item and IHSR-User. These results suggest that implicit hierarchical structures of users and items contain complementary information and capturing them simultaneously can further improve the recommendation performance.

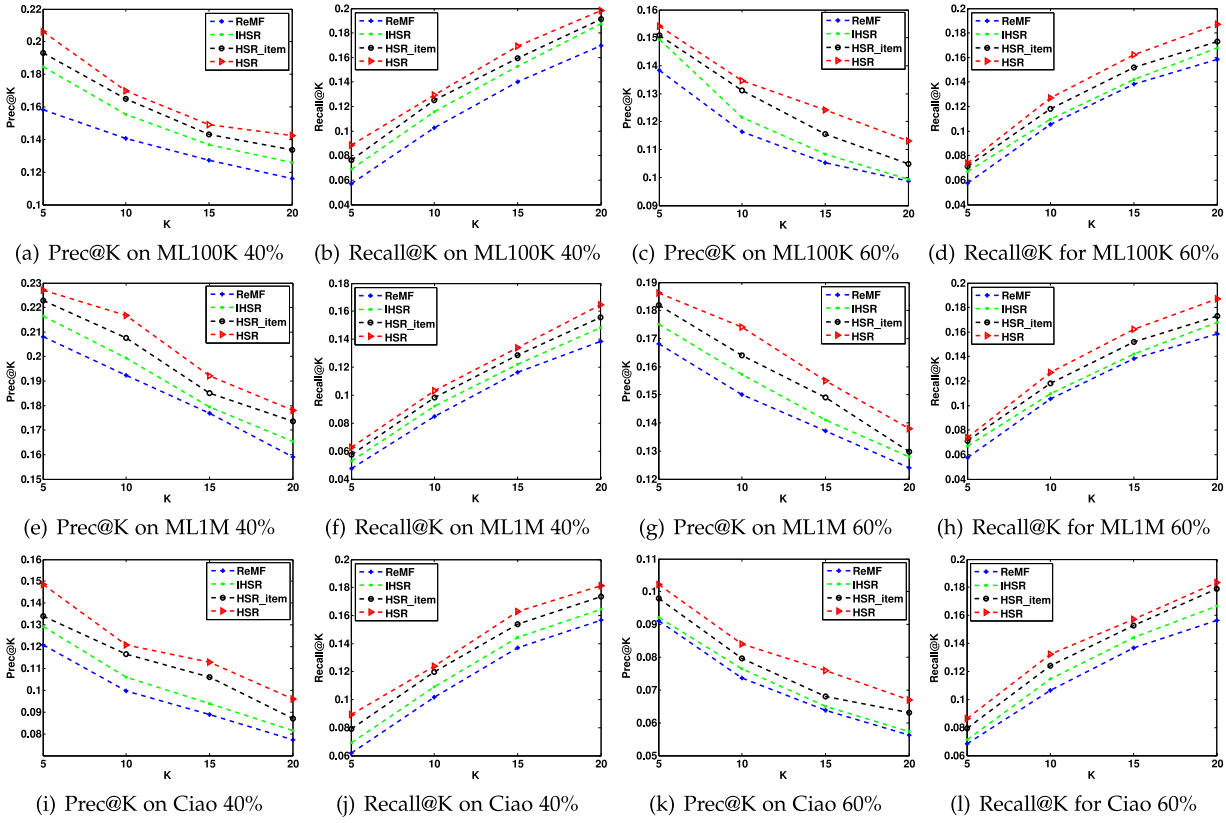
6.4 Performance Comparison of Recommender Systems with Hierarchical Structures

In this section, we conduct experiments to show the effectiveness of incorporating explicit hierarchical structures. The hierarchical structures are obtained by the category information of items. Since we don't have category information for Douban datasets, we only conduct experiments on MovieLens100K, MovieLens1M and Ciao. The representative compared methods are:

- ReMF: ReMF [18] is state-of-the-art model that exploits explicit hierarchical structures. Specifically, it is a matrix factorization framework with recursive regularization, which models the explicit hierarchical structure as a regularizer to guide the learning process user and item latent features. It doesn't work for implicit hierarchical structures.
- IHSR: The proposed model for exploring implicit hierarchical structure. This is used to compare the performance difference between implicit hierarchical structure and explicit hierarchical structure.
- HSR-Item: HSR-Item only considers the explicit item hierarchical structures, i.e., $\alpha = 0$ and $\beta > 0$. No explicit or implicit user preference structures are considered for this method.
- HSR: For HSR, we exploits implicit hierarchical structure of user preferences and explicit hierarchical of items. This setting is very practical in real-world as item hierarchical structures are usually explicitly available while explicit user preference structures are difficult to obtain.

Similarly, the parameters of all methods are determined via cross validation. Every experiment is conducted 10 times and the average performance for rating prediction and top-K recommendation are reported in Table 3 and Fig. 7, respectively. From the results, we make observe:

- Comparing HSR-Item and ReMF, HSR-Item is slightly better than ReMF. Both HSR-Item and ReMF utilize explicit hierarchical structure with the idea that the representation of child nodes should be close to that of parent node. However, HSR-Item is more effective as it uses deep non-negative matrix, which can learn more powerful latent features; and
- Generally, HSR-Item and IHSR has similar performance. This is because HSR-Item exploits item explicit hierarchical structures while IHSR explores

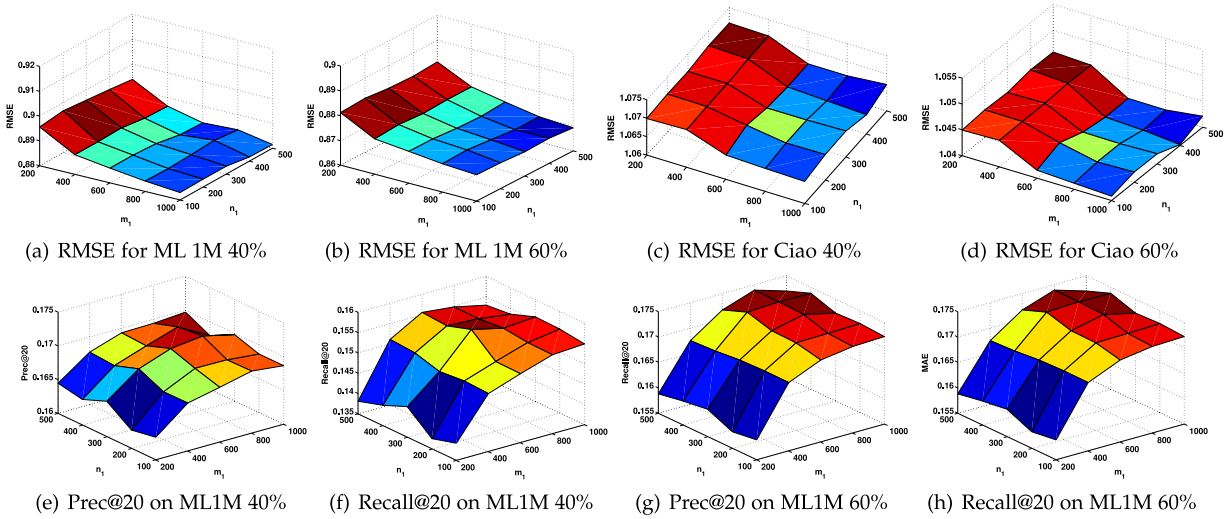


(a) Prec@K on ML100K 40% (b) Recall@K on ML100K 40% (c) Prec@K on ML100K 60% (d) Recall@K for ML100K 60%

(e) Prec@K on ML1M 40% (f) Recall@K on ML1M 40% (g) Prec@K on ML1M 60% (h) Recall@K for ML1M 60%

(i) Prec@K on Ciao 40% (j) Recall@K on Ciao 40% (k) Prec@K on Ciao 60% (l) Recall@K for Ciao 60%

Fig. 7. Top K ranking on MovieLens100K, Douban, MovieLens1M, and Ciao.



(a) RMSE for ML 1M 40% (b) RMSE for ML 1M 60% (c) RMSE for Ciao 40% (d) RMSE for Ciao 60%

(e) Prec@20 on ML1M 40% (f) Recall@20 on ML1M 40% (g) Prec@20 on ML1M 60% (h) Recall@20 on ML1M 60%

Fig. 8. Parameter analysis for IHSR.

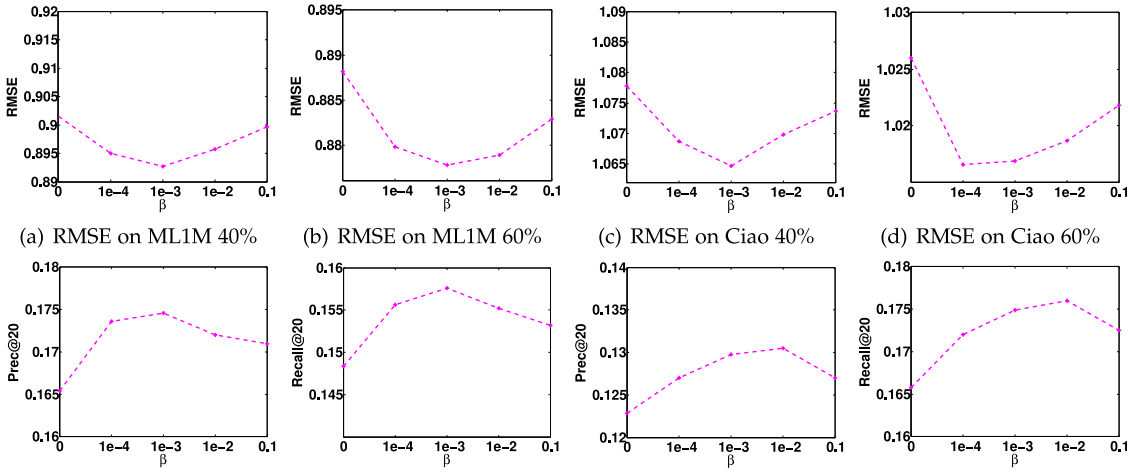
both user and item implicit hierarchical structures. HSR gives the best performance as in addition to item structures, it also exploits user preference hierarchical structures, which demonstrates the effectiveness of exploiting both explicit and implicit hierarchical structures for recommendation.

6.5 Parameter Analysis

In this section, we investigate the affects of the parameters on the performance of the proposed frameworks, IHSR and HSR. Specifically, for the proposed IHSR that explores implicit hierarchical structures, $m_1, \ldots, m_p$ and $n_1, \ldots, n_q$, which determines the number of nodes in each layer of implicit hierarchical structures, are important parameters to be selected; while for HSR that incorporates explicit hierarchical structures, $m_1, \ldots, m_p$ and $n_1, \ldots, n_q$ are given by the explicit HS. Instead, for HSR, α and β, which controls the contribution of the explicit hierarchical structures, are important parameters to be tuned. Since IHSR and HSR have different set of parameters that will affects their performance, next we will investigate the parameter sensitivity of IHSR and HSR, respectively.

6.5.1 Parameter Analysis for IHSR

In this section, we investigate the impact of dimensions of implicit layers on the performance of the proposed



(a) RMSE on ML1M 40% (b) RMSE on ML1M 60% (c) RMSE on Ciao 40% (d) RMSE on Ciao 60%

(e) Prec@20 on ML1M 40% (f) Recall@20 on ML1M 40% (g) Prec@20 on ML1M 60% (h) Recall@20 on ML1M 60%

Fig. 9. Parameter Analysis for HSR-Item.

framework IHSR. We only show results with $p = 2$ and $q = 2$, i.e., $\mathbf{W} \odot \mathbf{X} \approx \mathbf{W} \odot (\mathbf{U}_1\mathbf{U}_2\mathbf{V}_2\mathbf{V}_1)$ with $\mathbf{U}_1 \in \mathbb{R}^{n\times n_1}, \mathbf{U}_2 \in \mathbb{R}^{n_1\times d}, \mathbf{V}_1 \in \mathbb{R}^{d\times m_1}$, and $\mathbf{V}_2 \in \mathbb{R}^{m_1\times m}$ as we have similar observation with other settings of p and q. We fix d to be 20 and vary the value of n_1 as {100, 200, 300, 400, 500} and the value of m_1 as {200, 400, 600, 800, 1000}. Each experiments are conducted 10 times. The average performance for rating prediction on MovieLens 1M and Ciao and the average performance for top-K recommendation on MovieLens 1M are shown in Fig. 8. Form the figures, we make the following observation: (1) In general, when we increase the values of dimensions, the performance tends to first increase and then decrease; and (2) Among n_1 and m_1, the performance is relatively sensitive to m_1.

6.5.2 Parameter Analysis for HSR

In this section, we investigate the impact of β on the performance of the proposed framework HSR. Specifically, we evaluate the performance of HSR-Item as we only have the explicit hierarchical structures of items. We fix d to be 20 and vary the value of β as $\{0, 1e-4, 1e-3, 1e-2, 0.1, 1\}$. Similarity, we use x percent percent for training and $(100 - x)$ percent for testing where x is chosen as 40 and 60. Each experiments are conducted 10 times and the average performance for rating prediction on MovieLens 1M and Ciao and the average performance for top-K recommendation on MovieLens 1M are shown in Fig. 9. From the figure, we can observe that (i) Generally, as β increases, the performance first increases until certain value, then it decreases; and (ii) A value of β within $[10^{-4}, 10^{-3}]$ gives relatively good performance, which eases the parameter selection.

7 Conclusion

In this paper, we investigate the novel problem of exploiting implicit and explicit hierarchical structures of items and users for recommendation depends on their availability. We propose a novel recommendation framework, which integrates the explicit hierarchical structures of users into a coherent model when explicit hierarchical structures are available and exploits implicit hierarchical structures when the structures are not explicitly available. Experimental results on four real-world datasets demonstrate the importance of the explicit and implicit hierarchical structures of items and those of users in the recommendation performance improvement.

There are several interesting directions needing further investigation. First, in this work, we choose the weighted nonnegative matrix factorization as our basic model to capture the implicit hierarchical structures of items and users and we would like to investigate other basic models. Since social networks are pervasively available in social media and provide independent sources for recommendation, we will investigate how to incorporate social network information into the proposed framework.

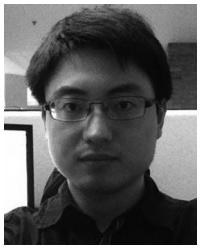
Acknowledgments

The work is supported by, or in part by, the US National Science Foundation (NSF) under the grant number #1614576, and the Office of Naval Research (ONR) under the grant number N00014-16-1-2257. This study is a significant extension of [35], which appeared in the *Proceedings of IJCAI 2015*.

References

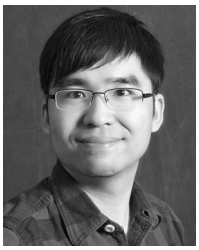
[1] P. Resnick and H. R. Varian, "Recommender systems," *Commun. ACM*, vol. 40, pp. 56–58, 1997.
[2] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Trans. Inf. Syst.*, vol. 22, pp. 89–115, 2004.
[3] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," in *Proc. 6th SIAM Int. Conf. Data Mining*, 2006, pp. 549–553.
[4] Y. Koren, "Collaborative filtering with temporal dynamics," *Commun. ACM*, vol. 53, no. 4, pp. 89–97, 2010.
[5] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1999, pp. 230–237.
[6] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H.-P. Kriegel, "Probabilistic memory-based collaborative filtering," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 1, pp. 56–69, Jan. 2004.
[7] J. Wang, A. P. De Vries, and M. J. Reinders, "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," in *Proc.29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2006, pp. 501–508.
[8] A. Mnih and R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. 20th Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 1257–1264.
[9] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Comput.*, vol. 42, no. 8, pp. 30–37, 2009.

[10] L. Si and R. Jin, "Flexible mixture model for collaborative filtering," in *Proc. 20th Int. Conf. Int. Conf. Mach. Learn.*, 2003, pp. 704–711.
[11] N. Srebro, J. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2004, pp. 1329–1336.
[12] Q. Gu, J. Zhou, and C. H. Ding, "Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs," in *Proc. Int. Conf. Data Mining*, 2010, pp. 199–210.
[13] J. Tang, X. Hu, H. Gao, and H. Liu, "Exploiting local and global social context for recommendation," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 2712–2718.
[14] J. M. Moreno-Jimenez and L. G. Vargas, "A probabilistic study of preference structures in the analytic hierarchy process with interval judgments," *Math. Comput. Modelling*, vol. 17, pp. 73–81, 1993.
[15] K. Lu, G. Zhang, R. Li, S. Zhang, and B. Wang, "Exploiting and exploring hierarchical structure in music recommendation," in *Information Retrieval Technology*. Berlin, Germany: Springer, 2012, pp. 211–225.
[16] M. Maleszka, B. Mianowska, and N. T. Nguyen, "A method for collaborative recommendation using knowledge integration tools and hierarchical structure of user profiles," *Knowl.-Based Syst.*, vol. 47, pp. 1–13, 2013.
[17] R. He, C. Lin, J. Wang, and J. McAuley, "Sherlock: Sparse hierarchical embeddings for visually-aware one-class collaborative filtering," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 3740–3746.
[18] J. Yang, Z. Sun, A. Bozzon, and J. Zhang, "Learning hierarchical feature influence for recommendation by recursive regularization," in *Proc. 10th ACM Conf. Recommender Syst.*, 2016, pp. 51–58.
[19] J. Tang, et al., "Recommendation with social dimensions," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 251–257.
[20] S. Wang, J. Tang, and H. Liu, "Toward dual roles of users in recommender systems," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manag.*, 2015, pp. 1651–1660.
[21] K. Shu, S. Wang, J. Tang, Y. Wang, and H. Liu, "Crossfire: Cross media joint friend and item recommendations," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2018.
[22] X. Meng, "Personalized privacy-preserving social recommendation," in *Proc. AAAI Conf. Artif. Intell.*, 2018.
[23] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, 2001, pp. 285–295.
[24] X. Ning, C. Desrosiers, and G. Karypis, "A comprehensive survey of neighborhood-based recommendation methods," in *Recommender Systems Handbook*. Berlin, Germany: Springer, 2015, pp. 37–76.
[25] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty in Artif. Intell.*, 2009, pp. 452–461.
[26] A. N. Nikolakopoulos, M. A. Kouneli, and J. D. Garofalakis, "Hierarchical itemspace rank: Exploiting hierarchy to alleviate sparsity in ranking-based recommendation," *Neurocomputing*, vol. 163, pp. 126–136, 2015.
[27] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proc. ACM Conf. Recommender Syst.*, 2010, pp. 39–46.
[28] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: Social recommendation using probabilistic matrix factorization," in *Proc. 17th ACM Conf. Inf. Knowl. Manage.*, 2008, pp. 931–940.
[29] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender Systems Handbook*. Berlin, Germany: Springer, 2015, pp. 191–226.
[30] J. Tang, C. Aggarwal, and H. Liu, "Recommendations in signed social networks," in *Proc. Int. Conf. World Wide Web*, 2016, pp. 31–40.
[31] S. Wang, Y. Wang, J. Tang, K. Shu, S. Ranganath, and H. Liu, "What your images reveal: Exploiting visual contents for point-of-interest recommendation," in *Proc. 6th Int. Conf. World Wide Web*, 2017, pp. 391–400.
[32] X. Meng, S. Wang, H. Liu, and Y. Zhang, "Exploiting emotion on reviews for recommender systems," in *Proc. AAAI Conf. Artif. Intell.*, 2018.
[33] J. Tang, X. Hu, and H. Liu, "Social recommendation: A review," *Social Netw. Anal. Mining*, vol. 3, pp. 1113–1133, 2013.
[34] A. Almahairi, K. Kastner, K. Cho, and A. Courville, "Learning distributed representations from reviews for collaborative filtering," in *Proc.9th ACM Conf. Recommender Syst.*, 2015, pp. 147–154.
[35] S. Wang, J. Tang, Y. Wang, and H. Liu, "Exploring implicit hierarchical structures for recommender systems," in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 1813–1819.
[36] Z. Sun, J. Yang, J. Zhang, and A. Bozzon, "Exploiting both vertical and horizontal dimensions of feature hierarchy for effective recommendation," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 189–195.
[37] S. E. Middleton, D. De Roure, and N. R. Shadbolt, "Ontology-based recommender systems," in *Handbook on Ontologies*. Berlin, Germany: Springer, 2009, pp. 779–796.
[38] F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding, "Community discovery using nonnegative matrix factorization," *Data Mining Knowl. Discovery*, vol. 22, no. 3, pp. 493–521, 2011.
[39] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Development Inf. Retrieval*, 2003, pp. 267–273.
[40] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. Schuller, "A deep semi-nmf model for learning hidden representations," in *Proc.31st Int. Conf. Int. Conf. Mach. Learn.*, 2014, pp. 1692–1700.
[41] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[42] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, 2011, pp. 287–296.
[43] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Conf. Neural Inf. Process. Syst.*, 2001, pp. 556–562.
[44] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 126–135.



Suhang Wang received the BS degree in electrical and computer engineering from Shanghai Jiao Tong University (SJTU), Shanghai, China, and the MS degree in EE:System from the University of Michigan, Ann Arbor, MI, in 2012 and 2013, respectively. He is currently working toward the PhD degree in computer science and engineering at Arizona State University (ASU), Tempe, AZ. He also works as a research assistant with the Data Mining and Machine Learning Lab (DMML) at ASU under the supervision of Dr. Huan Liu. His research interests include the recommender systems, feature selection, and representation learning. He has published innovative works in top conference proceedings such as WWW, AAAI, IJCAI, CIKM, SDM, WSDM, ICDM, and CVPR. He also worked as a research intern at IBM Research in 2016.

Jiliang Tang received the BS and MS degrees from the Beijing Institute of Technology, and the PhD degree in computer science from Arizona State University, in 2010 and 2015, respectively. He is an assistant professor of computer science engineering with Michigan State University. His research interests include social computing, data mining, and machine learning. He was awarded the Best Paper Award of SIGKDD 2016 and the Runner Up of the Best SIGKDD Dissertation Award 2015. He was among the organizers for various top conferences and serves as a regular journal reviewer and on numerous conference program committees. He has published innovative works in highly ranked journals and top conference proceedings that have received extensive coverage in the media.



Yilin Wang received the MS degree in computer science from Arizona State University, in 2013. He is currently working towards the PhD degree with the Arizona State University under the supervision of Professor Baoxin Li. His research interests include the data mining, recommendation system, computer vision, and deep learning. He has published innovative works in top conference proceedings such as WWW, AAAI, IJCAI, SDM WSDM, ICWSM, and CVPR. He also worked as a research intern in the Yahoo! Search Relevance Team at Yahoo Research in 2016.

Huan Liu received the BEng degree in computer science and electrical engineering from the Shanghai Jiao Tong University, and the PhD degree in computer science from the University of Southern California, Los Angeles, CA. He is currently a professor of computer science engineering with Arizona State University. Before he joined Arizona State University, he worked at Telecom Australia Research Labs and was on the faculty with the National University of Singapore. He was recognized for excellence in teaching and research in computer science and engineering at ASE. His research interests include the data mining, machine learning, social computing, and artificial intelligence, investigating problems that arise in many real-world, data-intensive applications with high-dimensional data of disparate forms such as social media. His well-cited publications include books, book chapters, encyclopedia entries as well as conference and journal papers. He serves on journal editorial boards and numerous conference program committees, and is a founding organizer of the International Conference Series on Social Computing, Behavioral-Cultural Modeling, and Prediction (http://sbp.asu.edu/). He is a fellow of the IEEE.