Exploring Implicit Hierarchical Structures for Recommender Systems

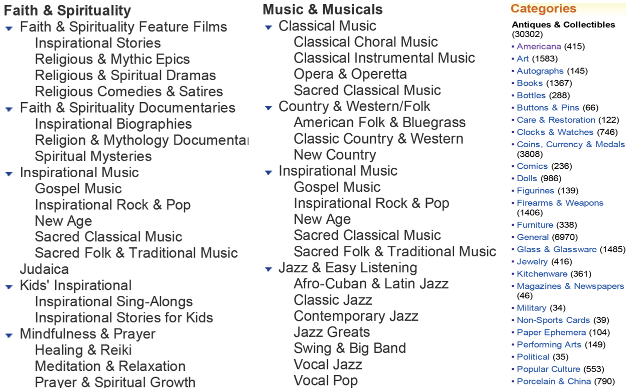
Suhang Wang, Jiliang Tang, Yilin Wang and Huan Liu
School of Computing, Informatics, and Decision Systems Engineering
Arizona State University, USA
{suhang.wang, jiliang.tang, yilin.wang.1, huan.liu}@asu.edu

Abstract

Items in real-world recommender systems exhibit certain hierarchical structures. Similarly, user preferences also present hierarchical structures. Recent studies show that incorporating the explicit hierarchical structures of items or user preferences can improve the performance of recommender systems. However, explicit hierarchical structures are usually unavailable, especially those of user preferences. Thus, there's a gap between the importance of hierarchical structures 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 HSR to bridge the gap, which enables us to capture the implicit hierarchical structures of users and items simultaneously. Experimental results on two real world datasets demonstrate the effectiveness of the proposed framework.

1 Introduction

Recommender systems [Resnick and Varian, 1997] 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 [Hofmann, 2004; Zhang *et al.*, 2006; Koren, 2010]. Collaborative Filtering can be roughly categorized into memory-based [Herlocker *et al.*, 1999; Yu *et al.*, 2004; Wang *et al.*, 2006] and model-based methods [Hofmann, 2004; Mnih and Salakhutdinov, 2007; Koren *et al.*, 2009]. 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, it has better performance than memory-based methods. Despite the success of various model-based methods [Si and Jin, 2003; Hofmann, 2004], matrix factorization (MF) based model has become one of the most popular methods due to its good performance and efficiency in handling large datasets[Srebro *et al.*, 2004; Mnih and Salakhutdinov, 2007; Koren *et al.*, 2009; Gu *et al.*, 2010; Tang *et al.*, 2013; Gao *et al.*, 2013].



Faith & Spirituality
- Faith & Spirituality Feature Films
  - Inspirational Stories
  - Religious & Mythic Epics
  - Religious & Spiritual Dramas
  - Religious Comedies & Satires
- Faith & Spirituality Documentaries
  - Inspirational Biographies
  - Religion & Mythology Documenta
  - Spiritual Mysteries
- Inspirational Music
  - Gospel Music
  - Inspirational Rock & Pop
  - New Age
  - Sacred Classical Music
  - Sacred Folk & Traditional Music
- Judaica
- Kids' Inspirational
  - Inspirational Sing-Alongs
  - Inspirational Stories for Kids
- Mindfulness & Prayer
  - Healing & Reiki
  - Meditation & Relaxation
  - Prayer & Spiritual Growth

(a) Faith & Spirituality

Music & Musicals
- Classical Music
  - Classical Choral Music
  - Classical Instrumental Music
  - Opera & Operetta
  - Sacred Classical Music
- Country & Western/Folk
  - American Folk & Bluegrass
  - Classic Country & Western
  - New Country
- Inspirational Music
  - Gospel Music
  - Inspirational Rock & Pop
  - New Age
  - Sacred Classical Music
  - Sacred Folk & Traditional Music
- Jazz & Easy Listening
  - Afro-Cuban & Latin Jazz
  - Classic Jazz
  - Contemporary Jazz
  - Jazz Greats
  - Swing & Big Band
  - Vocal Jazz
  - Vocal Pop

(b) Music & Musicals

Categories

Antiques & Collectibles (30302)
- Americana (415)
- Art (1583)
- Autographs (145)
- Books (1367)
- Bottles (288)
- Buttons & Pins (66)
- Care & Restoration (122)
- Clocks & Watches (746)
- Coins, Currency & Medals (3808)
- Comics (236)
- Dolls (986)
- Figurines (139)
- Firearms & Weapons (1406)
- Furniture (338)
- General (6970)
- Glass & Glassware (1485)
- Jewelry (416)
- Kitchenware (361)
- Magazines & Newspapers (46)
- Military (34)
- Non-Sports Cards (39)
- Paper Ephemera (104)
- Performing Arts (149)
- Political (35)
- Popular Culture (553)
- Porcelain & China (790)

(c) half.com

Figure 1: Netflix Movie Hierarchical Structure and half.com Book Hierarchical Structure

Items in real-world recommender systems could exhibit certain hierarchical structures. For example, Figure 1(a) and 1(b) 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. 1(a)). Similarly, Fig. 1(c) 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 [Moreno-Jimenez and Vargas, 1993]. For example, a user may generally prefer movies in *Faith Spirituality*, and more specifically, he/she watches movies under the sub-category of *Inspirational Stories*. Similarly, an antique clock collector may be interested in *Clocks & Watches* subcategory under the *Antiques & Collections* category. Items

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

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

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 [Lu *et al.*, 2012; Maleszka *et al.*, 2013]. Therefore, recently, there are recommender systems exploiting explicit hierarchical structures of items or users to improve recommendation performance [Lu *et al.*, 2012; Maleszka *et al.*, 2013]. However, explicit hierarchical structures are usually unavailable, especially those of users.

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 these two challenges, we propose a novel recommendation framework HSR, which captures implicit hierarchical structures of users and items based on the user-item matrix and integrate them into a coherent model. The major contributions of this paper are summarized next:

- We provide a principled approach to model implicit hierarchical structures of users and items simultaneously based on the user-item matrix;
- We propose a novel recommendation framework HSR, which enables us to capture implicit hierarchical structures of users and items when these structures are not explicitly available; and
- We conduct experiments on two 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 introduce the proposed framework HSR with the details of how to capture implicit hierarchical structures of users and items. In Section 3, we present a method to solve the optimization problem of HSR along with the convergence and time complexity analysis. In Section 4, we show empirical evaluation with discussion. In Section 5, we present the conclusion and future work.

2 The Proposed Framework

Throughout this 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 $\mathbf{X}(i,j) = 0$. 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.

2.1 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 [Zhang *et al.*, 2006]. 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:

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

where $\odot$ denotes Hadamard product and $\mathbf{W}(i,j)$ controls the contribution of $\mathbf{X}(i,j)$ to the learning process. A popular choice of $\mathbf{W}$ is - $\mathbf{W}(i,j) = 1$ if u_i rates v_j, and $\mathbf{W}(i,j) = 0$ otherwise.

2.2 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 [Wang *et al.*, 2011] and clusters of items [Xu *et al.*, 2003]. 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 subsection, we first give details about how to model implicit hierarchical structures based on weighted nonnegative matrix factorization, and then introduce the proposed framework HSR.

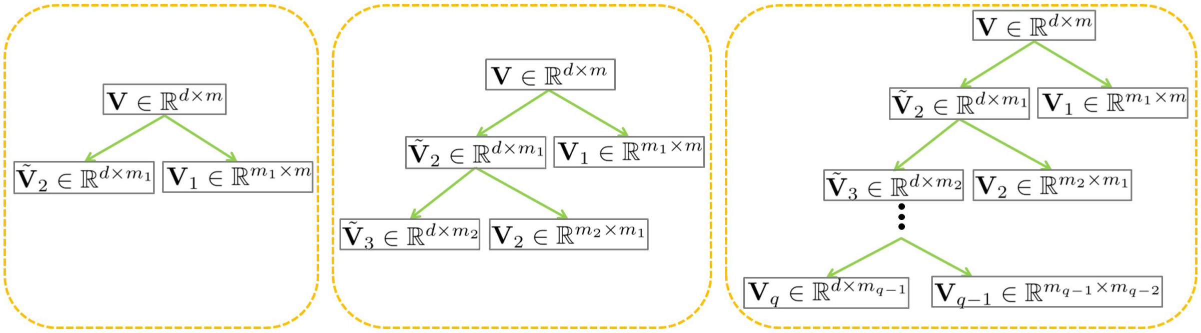
The item characteristic matrix $\mathbf{V} \in \mathbb{R}^{d\times m}$ indicates the affiliation of m items to d latent categories. Since $\mathbf{V}$ is nonnegative, we can further decompose $\mathbf{V}$ into two nonnegative matrices $\mathbf{V}_1 \in \mathbb{R}^{m_1\times m}$ and $\tilde{\mathbf{V}}_2 \in \mathbb{R}^{d\times m_1}$ to get a 2-layer implicit hierarchical structure of items as shown in Figure 2(a):

$$\mathbf{V} \approx \tilde{\mathbf{V}}_2\mathbf{V}_1 \quad (2)$$

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. We name $\tilde{\mathbf{V}}_2$ as the *latent category affiliation* matrix for the 2-layer implicit hierarchical structure because it indicates the affiliation relation between d latent categories in the 1-st layer and m_1 latent sub-categories in the 2-nd layer. Since $\tilde{\mathbf{V}}_2$ is non-negative, we can further decompose the latent category affiliation 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 Figure 2(b):

$$\mathbf{V} \approx \tilde{\mathbf{V}}_3\mathbf{V}_2\mathbf{V}_1 \quad (3)$$

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


$\mathbf{V} \in \mathbb{R}^{d \times m}$
$\tilde{\mathbf{V}}_2 \in \mathbb{R}^{d \times m_1}$
$\mathbf{V}_1 \in \mathbb{R}^{m_1 \times m}$
$\tilde{\mathbf{V}}_3 \in \mathbb{R}^{d \times m_2}$
$\mathbf{V}_2 \in \mathbb{R}^{m_2 \times m_1}$
$\mathbf{V}_q \in \mathbb{R}^{d \times m_{q-1}}$
$\mathbf{V}_{q-1} \in \mathbb{R}^{m_{q-1} \times m_{q-2}}$

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

Figure 2: Implicit Hierarchical Structures of Items via Deeply Factorizing the Item Characteristic Matrix.

structure by further factorizing $\tilde{\mathbf{V}}_{q-1}$ into two non-negative matrices as shown in Figure 2(c):

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

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

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

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.

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

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

An illustration of the proposed framework HSR is demonstrated in Figure 3. The proposed framework HSR 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 Figure 3.

3 An Optimization Method for HSR

The objective function in Eq.(6) 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 [Trigeorgis *et al.*, 2014] and then we will give convergence analysis and complexity analysis of the optimization method.

3.1 Inferring Parameters of HSR

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.(6) can be rewritten as:

$$\min_{\mathbf{U}_i \geq \mathbf{0}} ||\mathbf{W} \odot (\mathbf{X} - \mathbf{A}_i \mathbf{U}_i \mathbf{H}_i)||_F^2 + \lambda ||\mathbf{U}_i||_F^2 \tag{7}$$

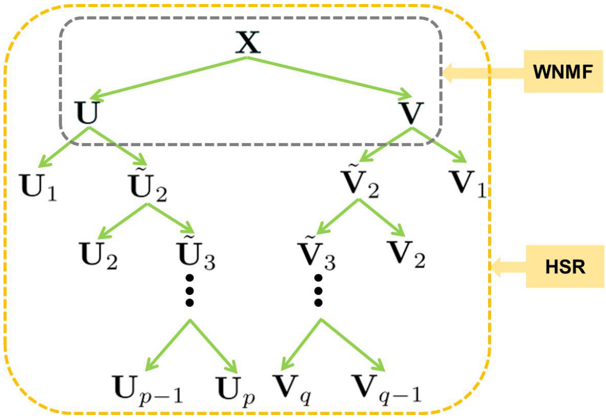


Figure 3: An Illustration of The Proposed Framework HSR.

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

$$\mathbf{A}_i = \begin{cases} \mathbf{U}_1 \mathbf{U}_2 \ldots \mathbf{U}_{i-1} & \text{if } i \neq 1 \\ \mathbf{I} & \text{if } i = 1 \end{cases} \tag{8}$$

And

$$\mathbf{H}_i = \begin{cases} \mathbf{U}_{i+1} \ldots \mathbf{U}_p \mathbf{V}_q \ldots \mathbf{V}_1 & \text{if } i \neq p \\ \mathbf{V}_q \ldots \mathbf{V}_1 & \text{if } i = p \end{cases} \tag{9}$$

The Lagrangian function of Eq.(7) is

$$\mathcal{L}(\mathbf{U}_i) = ||\mathbf{W} \odot (\mathbf{X} - \mathbf{A}_i \mathbf{U}_i \mathbf{H}_i)||_F^2 + \lambda ||\mathbf{U}_i||_F^2 - Tr(\mathbf{P}^T \mathbf{U}_i) \tag{10}$$

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

$$\frac{\partial \mathcal{L}(\mathbf{U}_i)}{\partial \mathbf{U}_i} = 2\mathbf{A}_i^T \left[\mathbf{W} \odot (\mathbf{A}_i \mathbf{U}_i \mathbf{H}_i - \mathbf{X})\right] \mathbf{H}_i^T + 2\lambda \mathbf{U}_i - \mathbf{P} \tag{11}$$

By setting the derivative to zero and using Karush-Kuhn-Tucker complementary condition [Boyd and Vandenberghe, 2004], i.e., $\mathbf{P}(s,t)\mathbf{U}_i(s,t) = 0$, we get:

$$\left[\mathbf{A}_i^T \left[\mathbf{W} \odot (\mathbf{A}_i \mathbf{U}_i \mathbf{V} - \mathbf{X})\right] \mathbf{H}_i^T + \lambda \mathbf{U}_i\right](s,t) \mathbf{U}_i(s,t) = 0 \tag{12}$$

Eq.(12) leads to the following update rule of $\mathbf{U}_i$ as:

$$\mathbf{U}_i(s,t) \leftarrow \mathbf{U}_i(s,t) \sqrt{\frac{\left[\mathbf{A}_i^T (\mathbf{W} \odot \mathbf{X}) \mathbf{H}_i^T\right](s,t)}{\left[\mathbf{A}_i^T (\mathbf{W} \odot (\mathbf{A}_i \mathbf{U}_i \mathbf{H}_i)) \mathbf{H}_i^T + \lambda \mathbf{U}_i\right](s,t)}} \tag{13}$$

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:

$$\min_{\mathbf{V}_i \geq \mathbf{0}} ||\mathbf{W} \odot (\mathbf{X} - \mathbf{B}_i \mathbf{V}_i \mathbf{M}_i)||_F^2 + \lambda ||\mathbf{V}_i||_F^2 \tag{14}$$

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

$$\mathbf{B}_i = \begin{cases} \mathbf{U}_1 \dots \mathbf{U}_p \mathbf{V}_q \dots \mathbf{V}_{i+1} & \text{if } i \neq q \\ \mathbf{U}_1 \dots \mathbf{U}_p & \text{if } i = q \end{cases} \tag{15}$$

and

$$\mathbf{M}_i = \begin{cases} \mathbf{V}_{i-1} \dots \mathbf{V}_1 & \text{if } i \neq 1 \\ \mathbf{I} & \text{if } i = 1 \end{cases} \tag{16}$$

We can follow a similar way as $\mathbf{U}_i$ to derive update rule for $\mathbf{V}_i$ as:

$$\mathbf{V}_i(s,t) \leftarrow \mathbf{V}_i(s,t) \sqrt{\frac{[\mathbf{B}_i^T (\mathbf{W} \odot \mathbf{X}) \mathbf{M}_i^T]\,(s,t)}{[\mathbf{B}_i^T (\mathbf{W} \odot (\mathbf{B}_i \mathbf{V}_i \mathbf{M}_i)) \mathbf{M}_i^T + \lambda \mathbf{V}_i]\,(s,t)}} \tag{17}$$

Algorithm 1 The Optimization Algorithm for the Proposed Framework HSR.

```
Input: $\mathbf{X} \in \mathbb{R}^{n \times m}, \lambda, p, q, d$ 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: $\tilde{\mathbf{U}}_1, \tilde{\mathbf{V}}_1 \leftarrow$ WNMF($\mathbf{X}, d$)
 3: for i = 1 to p-1 do
 4:     $\mathbf{U}_i, \tilde{\mathbf{U}}_{i+1} \leftarrow$ NMF($\tilde{\mathbf{U}}_i, n_i$)
 5: end for
 6: for i = 1 to q-1 do
 7:     $\tilde{\mathbf{V}}_{i+1}, \mathbf{V}_i \leftarrow$ 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:     for i = 1 to p do
12:         update $\mathbf{B}_i$ and $\mathbf{M}_i$ using Eq.(15) and Eq.(16)
13:         update $\mathbf{V}_i$ by Eq.(17)
14:     end for
15:
16:     for i = p to 1 do
17:         update $\mathbf{A}_i$ and $\mathbf{H}_i$ using Eq.(8) and Eq.(9)
18:         update $\mathbf{U}_i$ by Eq.(13)
19:     end for
20: until Stopping criterion is reached
21: predict rating matrix $\mathbf{X}_{pred} = \mathbf{U}_1 \dots \mathbf{U}_p \mathbf{V}_q \dots \mathbf{V}_1$
```

With the update rules for $\mathbf{U}_i$ and $\mathbf{V}_j$, the optimization algorithm for HSR is shown in Algorithm 3.1. Next we briefly review Algorithm 3.1. 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 pretraining, we first use WNMF [Zhang *et al.*, 2006] 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 3.1 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.(13) and Eq.(17) 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 3.1 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 $\mathbf{X}_{pred}(i,j)$[3].

3.2 Convergence Analysis

In this subsection, we will investigate the convergence of Algorithm 3.1. Following [Lee and Seung, 2001], we will use the auxiliary function approach to prove the convergence of the algorithm.

Definition [Lee and Seung, 2001] $G(h, h^{'})$ is an auxiliary function for $F(h)$ if the conditions

$$G(h, h^{'}) \geq F(h), G(h,h) = F(h) \tag{18}$$

are satisfied

Lemma 3.1 *[Lee and Seung, 2001] If G is an auxiliary function for F, then F is non-increasing under the update*

$$h^{(t+1)} = \arg\min G(h, h^{(}t)) \tag{19}$$

Proof $F(h^{t+1}) \leq G(h^{(t+1)}, h^{(t)}) \leq G(h^{(t)}, h^{(t)}) \leq G(h^{(t)})$

Lemma 3.2 *[Ding* et al.*, 2006] For any matrices $\mathbf{A} \in \mathbb{R}_+^{n \times n}, \mathbf{B} \in \mathbf{R}_+^{k \times k}, \mathbf{S} \in \mathbb{R}_+^{k \times k}, \mathbf{S}^{'} \in \mathbb{R}_+^{k \times k}$ and $\mathbf{A}, \mathbf{B}$ are symmetric, the following inequality holds*

$$\sum_{s=1}^{n} \sum_{t=1}^{k} \frac{(\mathbf{A}\mathbf{S}^{'}\mathbf{B})(s,t)\mathbf{S}^2(s,t)}{\mathbf{S}^{'}(s,t)} \geq Tr(\mathbf{S}^T \mathbf{A} \mathbf{S} \mathbf{B}) \tag{20}$$

Now consider the objective function in Eq.(7), it can be written in the following form by expanding the quadratic terms and removing terms that are irrelevant to $\mathbf{U}_i$

$$\begin{aligned} \mathcal{J}(\mathbf{U}_i) &= Tr\left(-2\mathbf{A}_i^T (\mathbf{W} \odot \mathbf{X}) \mathbf{H}_i^T \mathbf{U}_i^T\right) \\ &+ Tr\left(\mathbf{A}_i^T \left(\mathbf{W} \odot (\mathbf{A}_i^T \mathbf{U}_i \mathbf{H}_i)\right) \mathbf{H}_i^T \mathbf{U}_i^T\right) \\ &+ Tr(\lambda \mathbf{U}_i \mathbf{U}_i^T) \end{aligned} \tag{21}$$

Theorem 3.3 *The following function*

$$\begin{aligned} &\mathbf{G}(\mathbf{U}, \mathbf{U}^{'}) \\ &= -2 \sum_{s,t} (\mathbf{A}_i^T (\mathbf{W} \odot \mathbf{X}) \mathbf{H}_i^T)(s,t) \mathbf{U}_i(s,t) \left(1 + \log \frac{\mathbf{U}_i(s,t)}{\mathbf{U}_i^{'}(s,t)}\right) \\ &+ \sum_{s,t} \frac{(\mathbf{A}_i^T \left(\mathbf{W} \odot (\mathbf{A}_i^T \mathbf{U}_i \mathbf{H}_i)\right) \mathbf{H}_i^T)(s,t) \mathbf{U}_i^2(s,t)}{\mathbf{U}_i^{'}(s,t)} \\ &+ Tr(\lambda \mathbf{U}_i \mathbf{U}_i^T) \end{aligned} \tag{22}$$

[3] The code can be downloaded from http://www.public.asu.edu/∼swang187/

is an auxiliary function for $\mathcal{J}(\mathbf{U}_i)$. Furthermore, it is a convex function in $\mathbf{U}_i$ and its global minimum is

$$\mathbf{U}_i(s,t) \leftarrow \mathbf{U}_i(s,t)\sqrt{\frac{[\mathbf{A}_i^T(\mathbf{W}\odot\mathbf{X})\mathbf{H}_i^T]\,(s,t)}{[\mathbf{A}_i^T(\mathbf{W}\odot(\mathbf{A}_i\mathbf{U}_i\mathbf{H}_i))\mathbf{H}_i^T+\lambda\mathbf{U}_i]\,(s,t)}} \tag{23}$$

Proof The proof is similar to that in [Gu *et al.*, 2010] and thus we omit the details.

Theorem 3.4 *Updating $\mathbf{U}_i$ with Eq.(13) will monotonically decrease the value of the objective in Eq.(6).*

Proof With Lemma 3.1 and Theorem 3.3, we have $\mathcal{J}(\mathbf{U}_i^{(0)}) = G(\mathbf{U}_i^{(0)}, \mathbf{U}_i^{(0)}) \geq G(\mathbf{U}_i^{(1)}, \mathbf{U}_i^{(0)}) \geq \mathcal{J}(\mathbf{U}_i^{(1)}) \geq \ldots$. That is, $\mathcal{J}(\mathbf{U}_i)$ decreases monotonically.

Similarly, the update rule for $\mathbf{V}_i$ will also monotonically decrease the value of the objective in Eq.(6). Since the value of the objective in Eq.(6) is at least bounded by zero , we can conclude that the optimization method in Algorithm 3.1 converges.

3.3 Complexity Analysis

Initialization and fine-tuning are two most expensive operations for Algorithm 3.1. 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 $\mathcal{O}(tn_{i-1}n_id)$ for $1 < i < p$, and $\mathcal{O}(tnn_1d)$ for $i = 1$, where t is number of iterations takes for the decomposition. Thus the cost of initializing $\mathbf{U}_i$'s is $\mathcal{O}(td(nn_1 + n_1n_2 + \cdots + n_{p-2}n_{p-1})$. Similarly, the cost of initializing $\mathbf{V}_i$'s is $\mathcal{O}(td(mm_1 + m_1m_2 + \cdots + m_{q-2}m_{q-1})$ (line 6 to 8). The computational cost of fine-tuning $\mathbf{U}_i$ in each iteration is $\mathcal{O}(nn_{i-1}n_i + nn_im + n_{i-1}n_im)$. Similarly, the computational cost of fine-tuning $\mathbf{V}_i$ in each iteration is $\mathcal{O}(mm_{i-1}m_i + mm_in + m_{i-1}m_in)$. Let $n_0 = n, m_0 = m, n_p = m_q = d$, then the time comlexity of fine-tuning is $\mathcal{O}(t_f[(n+m)(\sum_{i=1}^{p} n_{i-1}n_i + \sum_{j=1}^{q} m_{i-1}m_i) + nm(\sum_{i=1}^{p} n_i + \sum_{j=1}^{q} m_j)])$, where t_f is the number of iterations takes to fine-tune. The overall time conplexity is the sum of the costs of initialization and fine-tuning.

4 Experimental Analysis

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

4.1 Datasets

The experiments are conducted on two publicly available benchmark datasets, i.e., MovieLens100K [4] and Douban [5]. MovieLens100K consists of 100,000 movie ratings of 943 users for 1682 movies. We filter 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 1371 users and 1967 movies. For both datasets, users can rate movies with scores from 1 to 5. The statistics of the two datasets are summarized in Table 1.

Table 1: Statistics of the Datasets

Dataset	# of users	# of items	# of ratings
MovieLens100K	943	1682	100,000
Douban	1371	1967	149,623

4.2 Evaluation Settings

Two widely used evaluation metrics, i.e., mean absolute error (MAE) and root mean square error (RMSE), are adopted to evaluate the rating prediction performance. Specifically, MAE is defined as

$$MAE = \frac{\sum_{(i,j)\in\mathcal{T}} |\mathbf{X}(i,j) - \tilde{\mathbf{X}}(i,j)|}{|\mathcal{T}|} \tag{24}$$

and RMSE is defined as

$$RMSE = \sqrt{\frac{\sum_{(i,j)\in\mathcal{T}} \left(\mathbf{X}(i,j) - \tilde{\mathbf{X}}(i,j)\right)^2}{|\mathcal{T}|}} \tag{25}$$

where in both metrics, $\mathcal{T}$ denotes the set of ratings we want to predict, $\mathbf{X}(i,j)$ denotes the rating user i gave to item j and $\tilde{\mathbf{X}}(i,j)$ denotes the predicted rating from u_i to v_j. We random select $x\%$ as training set and the remaining $1 - x\%$ as testing set where x is varied as $\{40, 60\}$ is this work. The random selection is carried out 10 times independently, and the average MAE and RMSE are reported. A smaller RMSE or MAE value means better performance. Note that previous work demonstrated that *small improvement in RMSE or MAE terms can have a significant impact on the quality of the top-few recommendation*[Koren, 2008].

4.3 Performance Comparison of Recommender Systems

The comparison results are summarized in Tables 2 and 3 for MAE and RMSE, respectively. The baseline methods in the table 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 to v_j. We use the cosine similarity measure 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 [Koren *et al.*, 2009].
- **WNMF**: weighted nonnegative matrix factorization tries to decompose the weighted rating matrix into two nonnegative matrices to minimize the reconstruction error [Zhang *et al.*, 2006]. In this work, we choose **WNMF** as the basic model of the proposed framework HSR.

[4] http://grouplens.org/datasets/movielens/

[5] http://dl.dropbox.com/u/17517913/Douban.zip

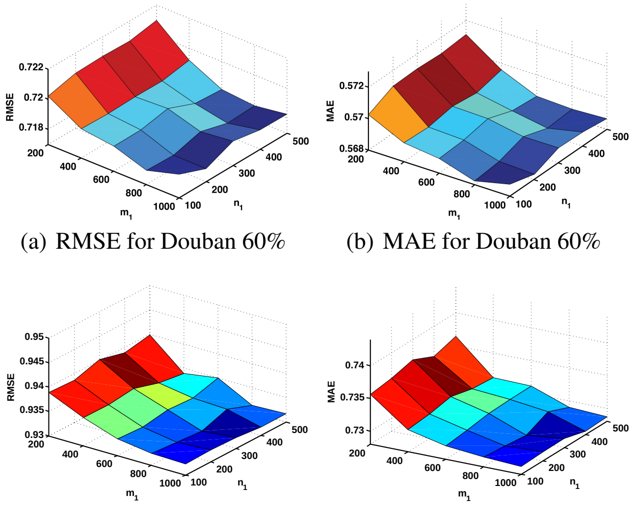
Table 2: MAE comparison on MovieLens100K and Douban

Methods		UCF	MF	WNMF	HSR-User	HSR-Item	HSR
MovieLens100K	40%	0.8392	0.7745	0.8103	0.7559	0.7551	0.7469
	60%	0.8268	0.7637	0.7820	0.7359	0.7363	0.7286
Douban	40%	0.6407	0.5973	0.6192	0.5792	0.5786	0.5767
	60%	0.6347	0.5867	0.6059	0.5726	0.5721	0.5685

Table 3: RMSE comparison on MovieLens100K and Douban

Methods		UCF	MF	WNMF	HSR-User	HSR-Item	HSR
MovieLens100K	40%	1.0615	0.9792	1.0205	0.9681	0.9672	0.9578
	60%	1.0446	0.9664	0.9953	0.9433	0.9412	0.9325
Douban	40%	0.8077	0.7538	0.7807	0.7313	0.7304	0.7284
	60%	0.7988	0.7403	0.7637	0.7225	0.7219	0.7179

- **HSR-Item**: HSR-Item is a variant of the proposed framework HSR. HSR-Item only considers the implicit hierarchical structure of items by setting $p = 1$ in HSR.
- **HSR-User**: HSR-User is a variant of the proposed framework HSR. HSR-Users only considers the implicit hierarchical structure of users by setting $q = 1$ in HSR.



(a) RMSE for Douban 60%

(b) MAE for Douban 60%

(c) RMSE for MovieLens100K 60%

(d) MAE for MovieLens100K 60%

Figure 4: Parameter Analysis for HSR.

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 [Koren *et al.*, 2009].
- Both **HSR-Item** and **HSR-Users** 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.
- **HSR** consistently outperforms both **HSR-Item** and **HSR-Users**. These results suggest that implicit hierarchical structures of users and items contain complementary information and capturing them simultaneously can further improve the recommendation performance.

4.4 Parameter Analysis

In this subsection, we investigate the impact of dimensions of implicit layers on the performance of the proposed framework HSR. 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 $\mathbf{V}_2 \in \mathbb{R}^{m_1 \times m}$, since we have similar observations 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 $\{200, 400, 600, 800, 1000\}$. We only show results with 60% of the datasets as training sets due to the page limitation and the results are shown in Figure 4. In general, when we increase the numbers of dimensions, the performance tends to first increase and then decrease. Among n_1 and m_1, the performance is relatively sensitive to m_1.

5 Conclusion

In this paper, we study the problem of exploiting the implicit hierarchical structures of items and users for recommendation when they are not explicitly available and propose a novel recommendation framework HSR, which captures the implicit hierarchical structures of items and users into a coherent model. Experimental results on two real-world datasets demonstrate the importance of the 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.

6 Acknowledgements

Suhang Wang, Jiliang Tang and Huan Liu are supported by, or in part by, the National Science Foundation (NSF) under grant number IIS-1217466 and the U.S. Army Research

Office (ARO) under contract/grant number 025071. Yilin Wang is supported by the NSF under contract/grant number 1135656. Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of the NSF and ARO.

References

[Boyd and Vandenberghe, 2004] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[Ding *et al.*, 2006] Chris Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135. ACM, 2006.

[Gao *et al.*, 2013] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 93–100. ACM, 2013.

[Gu *et al.*, 2010] Quanquan Gu, Jie Zhou, and Chris HQ Ding. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *SDM*, pages 199–210. SIAM, 2010.

[Herlocker *et al.*, 1999] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237. ACM, 1999.

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

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

[Koren, 2008] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[Koren, 2010] Yehuda Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

[Lee and Seung, 2001] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[Lu *et al.*, 2012] Kai Lu, Guanyuan Zhang, Rui Li, Shuai Zhang, and Bin Wang. Exploiting and exploring hierarchical structure in music recommendation. In *Information Retrieval Technology*, pages 211–225. Springer, 2012.

[Maleszka *et al.*, 2013] Marcin Maleszka, Bernadetta Mianowska, and Ngoc Thanh Nguyen. A method for collaborative recommendation using knowledge integration tools and hierarchical structure of user profiles. *Knowledge-Based Systems*, 47:1–13, 2013.

[Mnih and Salakhutdinov, 2007] Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2007.

[Moreno-Jimenez and Vargas, 1993] Jose Maria Moreno-Jimenez and Luis G Vargas. A probabilistic study of preference structures in the analytic hierarchy process with interval judgments. *Mathematical and Computer Modelling*, 17(4):73–81, 1993.

[Resnick and Varian, 1997] Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[Si and Jin, 2003] Luo Si and Rong Jin. Flexible mixture model for collaborative filtering. In *ICML*, volume 3, pages 704–711, 2003.

[Srebro *et al.*, 2004] Nathan Srebro, Jason Rennie, and Tommi S Jaakkola. Maximum-margin matrix factorization. In *Advances in neural information processing systems*, pages 1329–1336, 2004.

[Tang *et al.*, 2013] Jiliang Tang, Xia Hu, Huiji Gao, and Huan Liu. Exploiting local and global social context for recommendation. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2712–2718. AAAI Press, 2013.

[Trigeorgis *et al.*, 2014] George Trigeorgis, Konstantinos Bousmalis, Stefanos Zafeiriou, and Bjoern Schuller. A deep semi-nmf model for learning hidden representations. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1692–1700, 2014.

[Wang *et al.*, 2006] Jun Wang, Arjen P De Vries, and Marcel JT Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 501–508. ACM, 2006.

[Wang *et al.*, 2011] Fei Wang, Tao Li, Xin Wang, Shenghuo Zhu, and Chris Ding. Community discovery using nonnegative matrix factorization. *Data Mining and Knowledge Discovery*, 22(3):493–521, 2011.

[Xu *et al.*, 2003] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM, 2003.

[Yu *et al.*, 2004] Kai Yu, Anton Schwaighofer, Volker Tresp, Xiaowei Xu, and H-P Kriegel. Probabilistic memory-based collaborative filtering. *Knowledge and Data Engineering, IEEE Transactions on*, 16(1):56–69, 2004.

[Zhang *et al.*, 2006] Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. Learning from incomplete ratings using non-negative matrix factorization. In *SDM*, pages 549–553. SIAM, 2006.