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
{suhan.wang, jiliang.tang, yilin.wang, 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].

(a) Faith & Spirituality
(b) Music & Musicals
(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\footnote{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\footnote{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* sub-category under the *Antiques & Collections* category. Items

\footnote{1}Snapshots are from http://dvd.netflix.com/AllGenresList
\footnote{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 $\mathbf{B}$. 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 $\text{Tr}(\mathbf{M})$ 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 $\mathbf{X}(i,j) = \mathbf{U}(i,:)^T \mathbf{V}(:,j)$ by WNMF. $\mathbf{U}$ and $\mathbf{V}$ can be learned by solving the following optimization problem:

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

(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 non-negative, we can further decompose $\mathbf{V}$ into two nonnegative matrices $\tilde{\mathbf{V}}_1 \in \mathbb{R}^{m_1 \times m}$ 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):

$$\mathbf{V} \approx \tilde{\mathbf{V}}_2 \tilde{\mathbf{V}}_1$$

(2)

where $m_1$ is the number of latent sub-categories in the 2-nd layer and $\tilde{\mathbf{V}}_1$ 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 $\tilde{\mathbf{V}}_2$ into two nonnegative matrices $\tilde{\mathbf{V}}_2 \in \mathbb{R}^{m_2 \times m_1}$ and $\tilde{\mathbf{V}}_3 \in \mathbb{R}^{m \times m_2}$ to get a 3-layer implicit hierarchical structure of items as shown in Figure 2(b):

$$\mathbf{V} \approx \tilde{\mathbf{V}}_3 \tilde{\mathbf{V}}_2 \tilde{\mathbf{V}}_1$$

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

$$V \approx V_q V_{q-1} \ldots V_2 V_1$$  \hspace{1cm} (4)

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

$$U \approx U_1 U_2 \ldots U_{p-1} U_p$$  \hspace{1cm} (5)

where $U_1$ is a $n \times n_1$ matrix, $U_i \ (1 < i < p)$ is a $n_{i-1} \times n_i$ matrix and $U_p$ 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

$$\min_{U_1, \ldots, U_p, V_1, \ldots, V_q} \left\{ \| W \odot (X - U_1 \ldots U_p V_q \ldots V_1) \|_F^2 \\ + \lambda \sum_{i=1}^p \| U_i \|_F^2 + \sum_{j=1}^q \| V_j \|_F^2 \right\}$$  \hspace{1cm} s.t. U_i \geq 0, \ i \in \{1, 2, \ldots, p\}, \\
V_j \geq 0, \ j \in \{1, 2, \ldots, q\}$$  \hspace{1cm} (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 $U$ and the item characteristic matrix $V$ 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 $U_i$**

To update $U_i$, we fix the other variables except $U_i$. By removing terms that are irrelevant to $U_i$, Eq.(6) can be rewritten as:

$$\min_{U_i \geq 0} \| W \odot (X - A_i U_i H_i) \|_F^2 + \lambda \| U_i \|_F^2$$  \hspace{1cm} (7)

The Lagrangian function of Eq.(7) is

$$\mathcal{L}(U_i) = \| W \odot (X - A_i U_i H_i) \|_F^2 + \lambda \| U_i \|_F^2 - Tr(P^T U_i)$$  \hspace{1cm} (8)

where $A_i$ and $H_i, \ 1 \leq i \leq p$, are defined as:

$$A_i = \begin{cases} U_1 U_2 \ldots U_{i-1} & \text{if } i \neq 1 \\
I & \text{if } i = 1 \end{cases}$$  \hspace{1cm} (9)

$H_i$ is defined as:

$$H_i = \begin{cases} U_{i+1} \ldots U_p V_q \ldots V_1 & \text{if } i \neq p \\
V_q \ldots V_1 & \text{if } i = p \end{cases}$$  \hspace{1cm} (10)

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

$$A_i^T W (X - A_i U_i V - X) H_i^T + \lambda U_i^T (s,t) U_i(s,t) = 0$$  \hspace{1cm} (11)

Eq.(12) leads to the following update rule of $U_i$ as:

$$U_i(s,t) \leftarrow U_i(s,t) \sqrt{\frac{A_i^T W (X - A_i U_i H_i) H_i^T + \lambda U_i^T (s,t)}{A_i^T W (X - A_i U_i H_i) H_i^T}}$$  \hspace{1cm} (13)
Update Rule of $V_i$

Similarly, to update $V_i$, we fix the other variables except $V_i$. By removing terms that are irrelevant to $V_i$, the optimization problem for $V_i$ is:

$$
\min_{V_i \geq 0} \frac{1}{2} \| W \odot (X - B_i V_i M_i) \|^2_F + \lambda \| V_i \|^2_F
$$

(14)

where $B_i$ and $M_i$, $1 \leq i \leq q$, are defined as

$$
B_i = \begin{cases} 
U_1 \ldots U_p V_q \ldots V_{i+1} & \text{if } i \neq q \\
U_1 \ldots U_p & \text{if } i = q
\end{cases}
$$

(15)

and

$$
M_i = \begin{cases} 
V_{i-1} \ldots V_1 & \text{if } i \neq 1 \\
I & \text{if } i = 1
\end{cases}
$$

(16)

We can follow a similar way as $U_i$ to derive update rule for $V_i$ as:

$$
V_i(s,t) \leftarrow V_i(s,t) - \frac{[B_i^T (W \odot X) M_i^T]}{[B_i^T (W \odot (B_i V_i M_i)) M_i^T + \lambda V_i]} [s,t]
$$

(17)

**Algorithm 1** The Optimization Algorithm for the Proposed Framework HSR.

**Input:** $X \in \mathbb{R}^{m \times n}$, $\lambda, p, q, d$ and dimensions of each layer

**Output:** $X_{pred} = U_1 \ldots U_p V_q \ldots V_1$

1: Initialize $\{U_i\}_{i=1}^p$ and $\{V_i\}_{i=1}^q$
2: $U_1, V_1 \leftarrow$ WNMF($X, d$)
3: for $i = 1$ to $p-1$ do
4: $U_i, \hat{U}_{i+1} \leftarrow$ NMF($\hat{U}_i, n_i$)
5: end for
6: for $i = 1$ to $q-1$ do
7: $V_{i+1}, V_i \leftarrow$ NMF($\hat{V}_i, n_i$)
8: end for
9: $U_p = \hat{U}_p, V_q = \hat{V}_q$
10: repeat
11: for $i = 1$ to $p$ do
12: update $B_i$ and $M_i$ using Eq.(15) and Eq.(16)
13: update $V_i$ by Eq.(17)
14: end for
15: for $i = p$ to $1$ do
16: update $A_i$ and $H_i$ using Eq.(8) and Eq.(9)
17: update $U_i$ by Eq.(13)
18: end for
19: end for
20: until Stopping criterion is reached
21: predict rating matrix $X_{pred} = U_1 \ldots U_p V_q \ldots V_1$

With the update rules for $U_i$ and $V_i$, 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 $U_i$ and $V_i$. To perform pretraining, we first use WNMF [Zhang et al., 2006] to decompose the user-item rating matrix into $\hat{U}_1 \hat{V}_1$ by solving Eq.(1). After that, we further decompose $\hat{U}_1$ into $\hat{U}_1 \approx U_1 U_2$ and $\hat{V}_1 \approx V_2 V_1$ 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 $U_i$ and $V_i$ using updating rules in Eq.(13) and Eq.(17) separately. The procedure is to first update $V_i$ in sequence and then $U_i$ 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 $X_{pred} = U_1 \ldots U_p V_q \ldots V_1$. A missing rating from $u_i$ to $v_j$ will be predicted as $X_{pred}(i,j)$.

### 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

$$
G(h,h') \geq F(h), G(h,h) = F(h)
$$

(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

$$
h(t+1) = \arg \min G(h(t),h(t))
$$

(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 $A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{k \times k}, S \in \mathbb{R}^{n \times k}, S' \in \mathbb{R}^{k \times k}$ and $A,B$ are symmetric, the following inequality holds

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

(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 $U_i$

$$
\mathcal{J}(U_i) = Tr( -2A_i^T (W \odot X) H_i^T U_i^T ) + Tr( A_i^T (W \odot (A_i^T U_i H_i)) H_i^T U_i^T ) + Tr(\lambda U_i U_i^T)
$$

(21)

**Theorem 3.3** The following function

$$
G(U, U')
$$

$$
= -2 \sum_{s,t} \left( A_i^T (W \odot X) H_i^T \right)(s,t)U_i(s,t) \left( 1 + \log \frac{U_i(s,t)}{U_i'(s,t)} \right)
$$

$$
+ \frac{1}{2} \sum_{s,t} \left( A_i^T (W \odot (A_i^T U_i H_i)) H_i^T \right)(s,t)U_i(s,t)
$$

$$
+ Tr(\lambda U_i U_i^T)
$$

(22)

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

$$
U_i(s,t) \leftarrow U_i(s,t) \sqrt{ \frac{[A_i^T (W \odot X) H_i^T]}{[A_i^T (W \odot (A_i U_i H_i)) H_i^T + \lambda U_i]} [s,t]}
$$

(23)
Proof The proof is similar to that in [Gu et al., 2010] and thus we omit the details.

Theorem 3.4 Updating $U_i$ 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 $\mathcal{J}(U_i^{(0)}) = G(U_i^{(0)}, U^{(0)}_i) \geq G(U_i^{(1)}, U^{(0)}_i) \geq \mathcal{J}(U_i^{(1)}) \geq \ldots$. That is, $\mathcal{J}(U_i)$ decreases monotonically.

Similarly, the update rule for $V_i$ 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 $U_i \in \mathbb{R}^{n_i \times d}$ to $U_i \in \mathbb{R}^{n_i \times n}$, and $U_{i+1} \in \mathbb{R}^{n \times d}$ is $O((n_{i-1}n_i) + 1)$ for $1 < i < p$, and $O(tn_1n_d)$ for $i = 1$, where $t$ is the number of iterations taken for the decomposition. Thus the cost of initializing $U_i$’s is $O(td(n_1 + n_2 + \ldots + n_q - 1))$. Similarly, the cost of initializing $V_i$’s is $O(td(mm_1 + m_1m_2 + \cdots + m_qm_{q-1}))$ (line 6 to 8). The computational cost of fine-tuning $U_i$ in each iteration is $O((m_{i-1}n_i + n_mn + n_{i-1}n_m))$. Similarly, the computational cost of fine-tuning $V_i$ in each iteration is $O((mm_{i-1}m_i + mm_m + n_{i-1}m_i))$. Let $n_0 = n, m_0 = m, n_p = m_q = d$, then the time complexity of fine-tuning is $O(t_f((n + m)(\sum_{i=1}^{p-1}n_i - 1n_i + \sum_{j=1}^{q}m_{i-1}m_j) + mm(\sum_{i=1}^{p-1}n_i + \sum_{j=1}^{q}m_j)))$, where $t_f$ is the number of iterations taken to fine-tune. The overall time complexity 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 and Douban. 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

<table>
<thead>
<tr>
<th>Dataset</th>
<th># of users</th>
<th># of items</th>
<th># of ratings</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovieLens100K</td>
<td>943</td>
<td>1682</td>
<td>100,000</td>
</tr>
<tr>
<td>Douban</td>
<td>1371</td>
<td>1967</td>
<td>149,623</td>
</tr>
</tbody>
</table>

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

$$\text{MAE} = \frac{\sum_{(i,j) \in T} |X(i,j) - \hat{X}(i,j)|}{|T|}$$  \hspace{1cm} (24)

and RMSE is defined as

$$\text{RMSE} = \sqrt{\frac{\sum_{(i,j) \in T} (X(i,j) - \hat{X}(i,j))^2}{|T|}}$$  \hspace{1cm} (25)

where in both metrics, $T$ denotes the set of ratings we want to predict, $X(i,j)$ denotes the rating user $i$ gave to item $j$ and $\hat{X}(i,j)$ 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 $\{40, 60\}$ in 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.
- **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.

---

1) [http://grouplens.org/datasets/movielens/](http://grouplens.org/datasets/movielens/)
Table 2: MAE comparison on MovieLens100K and Douban

<table>
<thead>
<tr>
<th>Methods</th>
<th>UCF</th>
<th>MF</th>
<th>WNMF</th>
<th>HSR-User</th>
<th>HSR-Item</th>
<th>HSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovieLens100K</td>
<td>40%</td>
<td>0.8392</td>
<td>0.7745</td>
<td>0.8103</td>
<td>0.7559</td>
<td>0.7551</td>
</tr>
<tr>
<td></td>
<td>60%</td>
<td>0.8268</td>
<td>0.7637</td>
<td>0.7820</td>
<td>0.7359</td>
<td>0.7363</td>
</tr>
<tr>
<td>Douban</td>
<td>40%</td>
<td>0.6407</td>
<td>0.5973</td>
<td>0.6192</td>
<td>0.5792</td>
<td>0.5786</td>
</tr>
<tr>
<td></td>
<td>60%</td>
<td>0.6347</td>
<td>0.5867</td>
<td>0.6059</td>
<td>0.5726</td>
<td>0.5721</td>
</tr>
</tbody>
</table>

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., $W \odot X \approx W \odot (U_1U_2V_2V_1)$ with $U_1 \in \mathbb{R}^{n \times m_1}$, $U_2 \in \mathbb{R}^{m_1 \times d}$, $V_1 \in \mathbb{R}^{d \times m_2}$, and $V_2 \in \mathbb{R}^{m_2 \times n_1}$, 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 non-negative 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

This material is based upon work supported by, or in part by, the National Science Foundation (NSF) under grant number IIS-1217466, the U.S. Army Research Office (ARO) under contract/grant number 025071 and 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


