**Noname manuscript No.**
(will be inserted by the editor)

# Two-level Matrix Factorization for Recommender Systems

**Fangfang Li · Guandong Xu · Longbing Cao**

Received: date / Accepted: date

**Abstract** Many existing recommendation methods such as Matrix Factorization (MF) mainly rely on user-item rating matrix, which sometimes is not informative enough, often suffering from the cold start problem. To solve this challenge, complementary textual relations between items are incorporated into Recommender Systems (RS) in this paper. Specifically, we first apply a novel Weighted Textual Matrix Factorization (WTMF) approach to compute the semantic similarities between items, then integrate the inferred item semantic relations into MF and propose a Two-level Matrix Factorization (TLMF) model for RS. Experimental results on two open data sets not only demonstrate the superiority of TLMF model over benchmark methods, but also show the effectiveness of TLMF for solving the cold start problem.

**Keywords** Recommender System · Matrix Factorization · Latent Factor Model · Textual Semantic Relation

## 1 Introduction

Recommender Systems (RS) become increasingly important as they deeply involve our daily living, online, social, mobile and business activities. Typically, a set of users and items are involved, where each user $u$ rates various items according to his/her respective preferences (embodied by preference rates) [17]. A new rate or item is then recommended to a user based on the rating behaviors of similar users on existing items.

As one of Collaborative Filtering (CF) models, matrix factorization (MF) is a latent factor model [11] which predicts users' and items' overall structures in the form of latent factors. The preference of the active user to a specific item can be easily estimated by the multiplication of the decomposed user and item latent factors. How to

F. Li, G. Xu and L. Cao
University of Technology Sydney, Australia, 2007
Tel.: +61 4 5053 9811
E-mail: Fangfang.Li@student.uts.edu.au

correctly predict the latent factors of users and items is the key step of MF. Many researchers tried to incorporate side-information such as social relationships [16] [30], item relations [21] [6] [18] [1], topic distribution [2] [24] for more precisely predicting the latent factors of users and items. However, the existing MF algorithms have still not fully captured the intrinsic relations between items and users, especially the semantic relations. Therefore, we integrate the semantic relations between items into MF in this paper to improve the performance of RS. Specifically, we first deeply analyse items' semantic relations based on textual matrix factorization, then incorporate the semantic interactions into the upper level of MF on rating matrix.

Table 1: A Toy Example

| | $i_1$ | $i_2$ | $i_3$ | $i_4$ |
|---|---|---|---|---|
| $u_1$ | 1 | 3 | 5 | 4 |
| $u_2$ | 4 | 2 | 1 | 5 |
| $u_3$ | - | 2 | - | 4 |

| Item | Title | Introduction |
|---|---|---|
| $i_1$ | The Godfather | The aging patriarch of an organized crime dynasty ... |
| $i_2$ | Goodfellas | Henry Hill and his friends work their way up through ... |
| $i_3$ | Vertigo | A retired San Francisco detective investigates ... |
| $i_4$ | N or NW | Correspondence between young lovers nearly ... |

To illustrate the semantic relationships between items in RS, we give a toy example in Table 1. There is a rating matrix consisting of three users and four movies with textual information. Most existing CF methods utilize the rating matrix for recommendation but ignore the textual information of items. However, when the rating matrix is very sparse, the textual titles and introductions of items may also contribute to improve recommendation quality. Specifically, we can infer the movies' semantic relationships based on their textual introductions by text mining techniques. Intuitively, the existing text mining technique such as TF*IDF [28] can be applied to infer the relationships between items. However this method ignores the semantic relations between items. As we know that latent factor models such as Latent Semantic Analysis (LSA) [12], Probabilistic Latent Semantic Analysis (PLSA) [10], Latent Dirichlet Allocation (LDA) [4] are beneficial for identifying textual semantic relations. However, a fact that missing words of a sentence are irrelevant to the sentence was not considered in these latent factor models. Recently, a novel Weighted Textual Matrix Factorization (WTMF) method [8] was proposed to compute the semantic similarities between sentences and achieved better performance. However, this valuable semantic analysis had not been applied in RS, which greatly motivates us to model the textual semantic relations into MF approach of RS.

The contributions of the paper are summarized as follows:

- We apply the novel WTMF model to infer the semantic relations between items based on the textual information.
- We propose a Two-level Matrix Factorization (TLMF) recommendation model by accommodating item textual semantic relations and users' subjective rating preferences together.

– We conduct experiments to evaluate the superiority of the proposed TLMF model.

The rest of the paper is organized as follows. Section 2 presents the related work. In Section 3, we first state the notations in the paper, then give preliminary knowledge regarding matrix factorization in RS and text analysis. Section 4 analyses the semantic relations between items, which are then integrated into the proposed two-level matrix factorization model. Experimental results and analysis are presented in Section 5. The paper is concluded in the last Section.

## 2 Related Work

### 2.1 Recommendation Methods

Collaborative filtering (CF) is one of the most successful approaches taking advantage of user rating history to predict users' interests [23]. CF method mainly involves user-based CF and item-based CF. The basic idea of user-based CF is to recommend the interesting items to the active user according to the interests of the other users who have close relationships. Similarly, item-based CF tries to recommend the active user the potentially interested items having close similarities with the historical items that the active user likes. Although the wide adoption in many real applications, e.g., Amazon, the effect of CF is sharply decreased for new users and items. This is partly because when the rating matrix is very sparse, for new users and items, it is extremely difficult to get the relationships between users and between items. This limitation partly motivates us to consider other relations between users and between items, if we can get the users' or items' relationships no matter whether we have ample rating data, it may greatly enhance the effectiveness of recommendations. Actually, to some extent the semantic relations between items discussed in this paper can overcome the limitation.

As one of the most accurate single models for collaborative filtering, matrix factorization (MF) is a latent factor model [11] which effectively estimate latent factor vectors of users and items. Specifically, MF approach tries to decompose the rating matrix to user latent matrix and item latent matrix. Then the estimated rating is predicted by the multiplication of the two decomposed matrices. With the advent of social network, many researchers started to analyse social recommender systems and proposed various models integrating social networks [16] [30]. Social friendship is an outstanding explicit factor to improve the effectiveness of recommendation, however, not every web site has social or trust mechanisms. This explicit social gap strongly motivates us to explore other valuable relations between items and between users to improve recommendation qualities. Indeed, the discussed semantic relations between items in this paper are helpful for making reasonable recommendations over sparse rating matrix.

Content-based techniques [20] are another successful methods which recommend relevant items to users according to users' personal interests. Generally, attributes and free texts are two kinds of content in RS. Content-based methods often assume item's attributes are independent which is not always held in reality. Actually, several research outcomes such as [25] [26] [27] have been proposed to handle the challenging issues. Yu. [31] applied a coupled clustering method to group the items then exploited CF to make recommendations. Li. [13] [14] also applied item attributes to improve recommendation qualities. However, theses papers still did not consider the text information as complementary content, which greatly motivates us to consider textual semantic relations between items to enhance recommendation algorithms.

### 2.2 Textual Semantic Similarity

When textual information is considered in RS, text similarity is always a fundamental and important research. To date, many text similarity approaches are based on word similarity. However, most word similarity methods did not consider the semantic interaction between the word and its sentential context, which negatively impact the performance. It is well known that semantic analysis can help solve the issue of traditional word similarity. For example, latent factor models, such as Latent Semantic Analysis (LSA) [12], Probabilistic Latent Semantic Analysis (PLSA) [10], Latent Dirichlet Allocation (LDA) [4] can model the semantics of words and sentences simultaneously in the low-dimensional latent space. However, these methods did not model the missing words of a sentence which are helpful for restricting the relations between words and sentence. Recently, a Weighted Textual Matrix Factorization (WTMF) approach [8] was proposed to address this issue by modelling missing words and achieved better performance of semantic analysis. To the best of our knowledge, this valuable semantic analysis has not been applied in RS, which motivates us to model and incorporate the textual semantic relations into RS.

## 3 Preliminaries

### 3.1 Notations

A large number of user and item sets with textual information can be organized by a quadruple $S =< U, O, X, f >$, where $U = \{u_1, u_2, ..., u_n\}$ is a nonempty finite set of users, $O = \{o_1, o_2, ..., o_m\}$ is a nonempty finite set of items, $X$ is the term document matrix for item's textual information with $M$ terms and $N$ sentences. In the quadruple $S =< U, O, X, f >$, $f(u_i, o_j) = r_{ij}$ expresses the subjective rating preference on item $o_j$ for user $u_i$. Through the mapping function $f$, user rating preferences on items are then converted into a user-item matrix $R$, with $n$ rows and $m$ columns. Each element $r_{ij}$ of $R$ represents the rating given by user $u_i$ on item $o_j$. And the term document matrix $X$ derived from textual information of items can be decomposed into two latent factor matrices $A^{M\times K}$ and $B^{N\times K}$ which respectively represent word and sentence, where $K$ is the dimension of the latent factors for words and sentences, $M$ is the number of terms, and $N$ is the number of sentences in the textual corpus. Then the semantic similarities between items can be computed from the decomposed latent factor vectors of sentences, which are further incorporated into MF model of RS.

### 3.2 Matrix Factorization

Matrix factorization technique is widely employed in recommender systems. The goal of matrix factorization is to learn the latent preferences of users and the latent characteristics of items from all known ratings, then predict the unknown ratings through the inner products of user latent feature vectors and item latent feature vectors. Formally, matrix factorization based methods decompose the user-item rating matrix $R$ into two low rank latent feature matrices $P$ and $Q$. Then the matrix of predicted ratings $\hat{R} \in \mathbb{R}^{n\times m}$, where $n$, $m$ respectively denote the number of users and the number of items, can be modeled as:

$$\hat{R} = PQ^T = \begin{bmatrix} p_1 \\ p_2 \\ \dots \\ p_n \end{bmatrix} \begin{bmatrix} q_1 \ q_2 \ \dots \ q_m \end{bmatrix} \tag{1}$$

where, matrices $P \in \mathbb{R}^{n\times d}$ and $Q \in \mathbb{R}^{m\times d}$, $d$ is the rank (or dimension of the latent space) with $d \leq n, m$. The column vectors $p_u$ and $q_i$ represent the $d$-dimensional user-specific latent feature vector and item specific latent feature vector, respectively. Through this modelling, the prediction task of matrix $\hat{R}$ is transferred to compute the mapping of users and items to factor matrices $P$ and $Q$. Once this mapping is completed, the inner product of $p_u$ and $q_i$ can be easily utilized to predict the rating given by the active user $u$ for target item $o_i$.

In order to learn the optimum latent factor vectors of users and items, one way is to optimize the objective function 2 as Singular Value Decomposition (SVD) [29],

$$\frac{1}{2}\left\|R - PQ^T\right\|_F^2 \tag{2}$$

where $\|.\|_F^2$ is the Frobenius norm. But this optimization method cannot effectively determine the latent semantic factors for a very sparse rating matrix. Hence, another direct optimization method to factorize the observed ratings using the following objective function:

$$L = \min_{P,Q} \frac{1}{2} \sum_{(u,i)\in C} \left(R_{u,i} - p_u q_i^T\right)^2 + \frac{\lambda}{2}\left(\|P\|_F^2 + \|Q\|_F^2\right) \tag{3}$$

where $C$ indicates the set of the $(u; o_i)$ pairs for known ratings. To avoid over-fitting, two regularization terms on the sizes of $P$ and $Q$ are added in Eqn. 3 as constraints, and $\lambda$ represents the regularization parameters impacting on latent semantic vectors.

To learn the latent factor matrices $P$ and $Q$, an efficient stochastic gradient descent algorithm is often applied to optimize the objective function given in Equation 3 with an iteration process. The derivative of $L$ with respect to $p_u$ and $q_i$ are as follows:

$$\frac{\partial L}{\partial p_u} = \sum_{o_i} I_{u,o_i}(p_u {q_i}^T - R_{u,o_i})q_i + \lambda p_u \tag{4}$$

$$\frac{\partial L}{\partial q_i} = \sum_{u} I_{u,o_i}(p_u {q_i}^T - R_{u,o_i})p_u + \lambda q_i \tag{5}$$

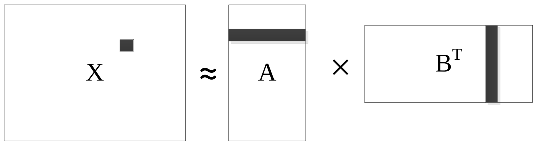


Fig. 1: Weighted Textual Matrix Factorization

Furthermore, the updating rules for iteration are derived to learn the latent vectors $p_u$ and $q_i$:

$$p_u \leftarrow p_u + \eta((R_{ui} - p_u {q_i}^T)q_i - \lambda p_u) \tag{6}$$

$$q_i \leftarrow q_i + \eta((R_{ui} - p_u {q_i}^T)p_u - \lambda q_i) \tag{7}$$

### 3.3 Weighted Textual Matrix Factorization

The Weighted Textual Matrix Factorization (WTMF) [8] has been successfully applied in many natural language processing tasks such as sentence similarity computation [7] and linkage analysis between tweets and news [9], achieving state-of-the-art unsupervised performance. A big contribution of WTMF is that the missing words are modelled to greatly enrich the features of sentences and short text. According to Guo and Diab, the missing words of a sentence are defined as all the vocabulary of the training corpus minus the observed words in a sentence. Intuitively, we know that a sentence should not be related to the missing words of the sentence (we call it missing words principle), which is often ignored by most sentence similarity measures. WTMF thus models sentences by textual MF method which considers missing words as additional constraints for the semantics of sentences.

Similar to the above traditional MF based on rating matrix in RS, textual MF is applied on the term document matrix. Specifically, the textual corpus is first represented by a term document matrix $X$ with the TF*IDF values of words in each cell, where the rows of $X$ are words and columns are sentences. Matrix $X$ is then similarly approximated by the product of a $M \times K$ matrix $A$ and a $N \times K$ matrix $B$. Accordingly, each sentence $s_j$ is represented by a $K$ dimensional latent factor vector $B_j$ , and each word $w_i$ is generalized by latent factor vector $A_i$. Therefore, the inner product of a word vector $A_i$ and a sentence vector $B_j$ is to approximate the cell $X_{ij}$ (shaded part in Fig. 1). Thereby, the constraint of the inner product of $A_i$ and $B_j$ to be close to 0 ensures the missing words are modelled in line with the above principle.

To overcome the over influence of missing words, a small weight $w_m$ is assigned for 0 cells in matrix $X$, since 99% of the cells are 0 values, possibly diminishing the effect of observed words. The model parameters (vectors in $A$ and $B$) are optimized by minimizing the objective function:

$$\begin{aligned} &L_l = \min_{A,B} \frac{1}{2} \sum_i \sum_j W_{ij} \left( X_{i,j} - A_i B_j^T \right)^2 + \frac{\gamma}{2} \left( \|A\|_F^2 + \|B\|_F^2 \right) \\ &W_{ij} = \begin{cases} 1, & if\ X_{ij} \neq 0 \\ w_m, & if\ X_{ij} = 0 \end{cases} \end{aligned} \tag{8}$$

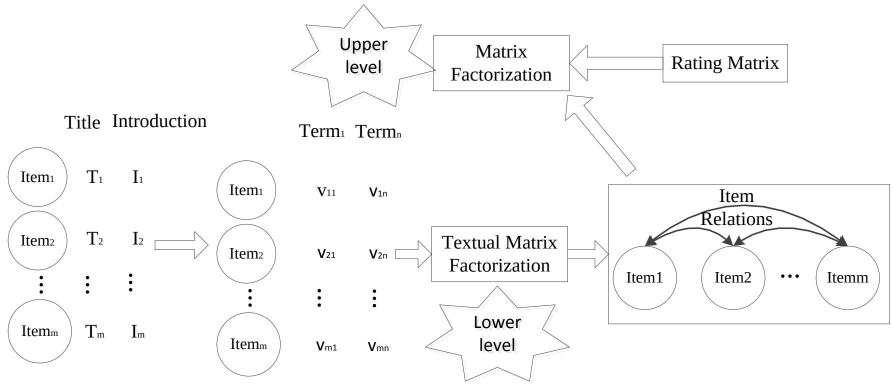


Fig. 2: Two-level Matrix Factorization Model

where $\gamma$ is a free regularization factor, and the weight matrix $W$ defines a weight for each cell in $X$.

## 4 Two-level Matrix Factorization

In this section, we mainly introduce the two-level MF method as shown in Fig. 2. TLMF first computes the semantic relations between items according to textual matrix factorization. Then items' semantic relations and users' rating preferences are incorporated together into TLMF model. The TLMF model is advantageous in the following aspects: (1) the semantic relations between items are analysed by the lower level MF, which are able to remedy the problem of lacking informative rating knowledge, further to improve the quality of recommendation, (2) user's subjective rating preference is also incorporated in the learning model by the upper level MF. To help understand the proposed model, we below respectively introduce items' semantic relations and how to integrate them to form TLMF model.

### 4.1 Semantic Analysis for Items

The semantic relations between items are derived from their textual information by the mentioned WTMF approach. According to Eqn. 8, the best latent factor matrices $A$ and $B$ need to be optimized first. Then the semantic relations between items $o_i$ and $o_j$ can be computed from that between sentences $s_i$ and $s_j$ by cosine similarity of vectors $B_i$ and $B_j$, which is given in Eqn. 9.

$$S(o_i, o_j) = S(s_i, s_j) = cos < B_i, B_j >= \frac{B_i.B_j}{\|B_i\|\|B_j\|} \tag{9}$$

To optimize the latent vectors, $A$ and $B$ are first randomly initialized, then can be computed iteratively by the following equations [19]:

$$A_i = (B\tilde{W}^{(i)}B^T + \lambda I)^{-1}B\tilde{W}^{(i)}X_{i,\cdot}^T \tag{10}$$

$$B_j = (A\tilde{W}^{(j)}A^T + \lambda I)^{-1}A\tilde{W}^{(i)}X_{\cdot,j}^T \tag{11}$$

where $A_i$ is a $K$-dimension latent semantic vector profile for word $w_i$; similarly, $B_j$ is the $K$-dimension vector profile for sentence $s_j$.

### 4.2 Two-level MF Model

MF approaches have been recognized as the main stream in RS through a latent topic projection learning model. In this work, we attempt to incorporate the discussed semantic relations between items into a MF scheme.

As shown in Fig. 2, TLMF not only takes the rating matrix, but also the relations between items into account. All these aspects should be accommodated into a unified learning model. The learning procedure is constrained by three-fold: the learned rating values should be as close as possible to the observed rating values, the predicted item profiles should be similar to their neighbourhoods as well, which are derived from their semantic relations. Besides, the lower level textual MF models relations between words and sentences by accommodating the impact of missing words, which improves the performance of semantic analysis. Specifically, in order to incorporate the relations between items, we add two additional regularization factors in the optimization step. Then the computation of the mapping can be similarly optimized by minimizing the regularized squared error. The objective function is given as Eqn. 12.

$$\begin{aligned} L_u = \frac{1}{2}\sum_{(u,o_i)\in K}\left(R_{u,o_i} - \hat{R}_{u,o_i}\right)^2 + \frac{\lambda}{2}\left(\|Q\|_F^2 + \|P\|_F^2\right) + \\ \frac{\alpha}{2}\sum_{o_i}\left(q_i - \sum_{o_j} S(o_i,o_j)q_j\right)\left(q_i - \sum_{o_j} S(o_i,o_j)q_j\right)^T \end{aligned} \tag{12}$$

In the objective function, the semantic relations between items and users' rating preferences are integrated together. This means the users' rating preferences and items' semantic relations act together to make recommendations. In addition, another distinct advantage is that, when we do not have ample rating data, it is still possible to make satisfactory recommendations via leveraging the semantic relations between items. Similar to Eqn.4 and 5, we optimize the above objective function by minimizing $L_u$ through the gradient decent approach:

$$\frac{\partial L_u}{\partial p_u} = \sum_{o_i} I_{u,o_i}(p_u {q_i}^T - R_{u,o_i})q_i + \lambda p_u \tag{13}$$

$$
\begin{aligned}
&\frac{\partial L_u}{\partial q_i} = \sum_u I_{u,o_i}(p_u {q_i}^T - R_{u,o_i})p_u + \lambda q_i + \\
&\alpha(q_i - \sum_{o_j \in \mathbb{N}(o_i)} S(o_i, o_j) q_j) - \\
&\alpha \sum_{o_j} S(o_j, o_i)(q_j - \sum_{o_k \in \mathbb{N}(o_j)} S(o_j, o_k) q_k)
\end{aligned}
\tag{14}
$$

where $I_{u,o_i}$ is an logical function indicating whether the user has rated item $o_i$ or not. $S(o_i, o_j)$ is the semantic similarity between items $o_i$ and $o_j$. $\mathbb{N}(o_i)$ represents the item neighborhood.

Furthermore, the updating rules for iteration are derived to learn the latent vectors $p_u$ and $q_i$:

$$p_u \leftarrow p_u + \eta((R_{ui} - p_u {q_i}^T) q_i - \lambda p_u) \tag{15}$$

$$
\begin{aligned}
&q_i \leftarrow q_i + \eta((R_{ui} - p_u {q_i}^T) p_u - \lambda q_i - \\
&\alpha(q_i - \sum_{o_j \in \mathbb{N}(o_i)} S(o_i, o_j) q_j) + \\
&\alpha \sum_{o_j} S(o_j, o_i)(q_j - \sum_{o_k \in \mathbb{N}(o_j)} S(o_j, o_k) q_k))
\end{aligned}
\tag{16}
$$

The optimum matrices $P$ and $Q$ can be computed by the above gradient descent approach. Generally, the TLMF model starts by computing item relations based on the textual content by the lower level TMF method, then commences an iteration process for optimizing $P$ and $Q$ until convergence, according to Eqn. 15 and 16. Once $P$ and $Q$ are learned, the ratings for user-item pairs $(u, o_i)$ can be easily predicted by Eqn. 1. Overall, the TLMF computation process can be described by algorithm 1.

### 4.3 Complexity Analysis

The main computation cost of the TLMF mainly involves learning the latent factor vectors and computing the similarity between items with the textual MF method. For learning the latent factor vectors, the main time cost is to evaluate the objective function $L_u$ and the corresponding gradients for users and items. The computational complexity of evaluating objective function is $O(m\bar{r}d + m\bar{t}d)$, where $\bar{r}$ is the average number of ratings per item and $\bar{t}$ is the average number of most similar neighbors per item. The time complexities of evaluating the latent factor vectors for users and items are $O(n\bar{x}d)$ and $O(m\bar{x}d + m\bar{t}^2 d)$, respectively, where $\bar{x}$ is the average number of ratings per user. We know that the value of $\bar{r}$ and $\bar{t}$ are usually small because the user-item rating matrix $R$ is sparse, and only the most similar neighbors are selected for the target item. Therefore, the computation of $L_u$ and latent factor vectors are fast and linear with respect to the number of items $m$ and users $n$ in the user-item rating matrix $R$.

For computing the textual similarity by the textual MF, the main time cost is to learn the latent factor vectors $A$ and $B$ for textual terms and sentences. Similar to the complexity analysis for the upper level MF, the computation complexity for

**Algorithm 1:** Two-level Matrix Factorization Algorithm

**Input**: $R$: the user-item rating matrix.
$d$: the dimension of latent feature vector on upper level.
$K$: the dimension of latent feature vector on lower level.
$T$: the textual corpus of items.
$Z$: the number of iterations.
$\lambda$: the regularization parameter for upper level MF.
$\gamma$: the regularization parameter for lower level MF.
$\eta$: the learning rate.
$w_m$: the parameter of missing words.
**Output**: P: the user latent feature matrix.
Q: the item latent feature matrix.

Build term document matrix with TF*IDF values based on textual corpus of items $T$
Initiate $A^0$ and $B^0$ with random decimals and $j$=0
**while** $j < Z$ or ($L_l^j - L_l^{j+1} \leq \epsilon$) **do**
  **for** all words and sentences
    Compute and update $A$ and $B$ by Eqn. 10 and 11
  **end for**
  $j$++
**end while**
Compute similarity matrix $S$ by Eqn. 9
Initiate $P^0$ and $Q^0$ with random decimals and $j$=0
**while** $j < Z$ or ($L_u^j - L_u^{j+1} \leq \epsilon$) **do**
  **for** all users and items
    Compute and update $P$ and $Q$ by Eqn. 13 to 16
  **end for**
  $j$++
**end while**
**return** $P$ and $Q$

computing objective function $L_l$ is $O(NaK)$, where $N$ is the number of sentences in the textual corpus, $K$ is the dimension of the latent vectors, $a$ is the average number of terms per sentence. The time complexity for evaluating the latent factor vectors $A$ and $B$ are respectively $O(MaK)$ and $O(NaK)$. Therefore, the evaluation of the lower level MF is dependent on the number of terms for sentences which is impacted by the sentence length. That means the longer of the textual sentences the more time consuming for inferring the similarities between items. Fortunately, the computation process of textual similarity between items can be offline, which can also be very beneficial for the recommendation community. Online recommendation strategy can be possibly implemented on big data platforms such as Hadoop or Spark.

## 5 Experiments and Results

In this section, we evaluate our proposed model and compare it with the existing approaches respectively using MovieLens[1] and BookCrossing[2] data sets.

[1] www.movielens.org
[2] www.bookcrossing.com

Table 2: Basic Statistics for Data Sets

| Statistics | MovieLens1m | BookCrossing |
|---|---|---|
| Num. of Ratings | 1,000,209 | 1,149,780 |
| Num. of Users | 6040 | 278,858 |
| Num. of Items | 3076 | 271,379 |
| Sparsity | 94.62% | 99.99% |

### 5.1 Data Sets

The MovieLens data set has been widely explored in RS research in the last decade. The MovieLens 1M data set consists of 1,000,209 anonymous ratings of approximately 3,076 movies made by 6,040 MovieLens users who joined MovieLens in 2000. The ratings are made on a 5-star scale and each user has at least 20 ratings. The movies have titles provided by the IMDB (including year of release), we also extract relevant descriptions about the movies from wikipedia.

Similarly, collected by Cai-Nicolas Ziegler from the Book-Crossing community, the Book-Crossing data set involves 278,858 users with demographic information providing 1,149,780 ratings on 271,379 books. The ratings range from 1 to 10 and the books' titles are used to form the item semantic relations. The basic statistics of MovieLens and BookCrossing data sets are shown in Table 2.

### 5.2 Experimental Settings

We perform 5-fold cross validation in our experiments for both MovieLens 1M and Book-Crossing data sets. We first split the original data into five equal sized samples, then we respectively keep a single sample of the five samples as the testing set, and the remaining four samples are used as training set. In this way, the original data is converted to a 5-fold data set with each fold 80% as the training set and 20% as the test set. Then the cross-validation process is repeated five times on the sampled data for each fold. Finally, the estimation on the whole data can be averaged from the five results for each fold. Here we use Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) as evaluation metrics, which are defined as follow:

$$RMSE = \sqrt{\frac{\sum_{(u,i)|R_{test}} \left(r_{u,i} - \hat{r}_{u,i}\right)^2}{|R_{test}|}} \tag{17}$$

$$MAE = \frac{\sum_{(u,i)|R_{test}} |r_{u,i} - \hat{r}_{u,i}|}{|R_{test}|} \tag{18}$$

where $R_{test}$ is the set of all pairs $(u, o_i)$ in the test set.

To evaluate the performance of our proposed TLMF we consider eight baseline approaches:

- The basic probabilistic matrix factorization (PMF) approach [22];
- Singular value decomposition (RSVD) [3] is a factorization method to decompose the rating matrix;

- Implicit social matrix factorization (ISMF) [15] is an unified model which incorporates implicit social relationships between users and between items computed by Pearson similarity based on the user-item rating matrix;
- User-based CF (UBCF) [23] first computes users' similarity by Pearson Correlation on the rating matrix, then recommends relevant items to the given user according to the users who have strong relationships;
- Item-based CF (IBCF) [5] first considers items' similarity by Pearson Correlation on the rating matrix, then recommends relevant items which have strong relationships with the given user's interested items
- MF model with edit distance (EDMF) applies edit distance measure to compute the similarities between items based on textual information, with the incorporation of item relations into MF as objective function shown in Enq. 12
- MF model with term frequency (TFMF) first directly computes the similarities between items using term frequency vectors after transforming items' textual information into a term-document matrix with the value of term frequency, then integrates the item relations into MF as Enq. 12;
- MF model with TFIDF (TIMF) directly calculates the similarities between items using term TFIDF vectors, followed by the same integration approach with TFMF.

## 5.3 Experiments and Discussions

We respectively evaluate the effectiveness of our TLMF model in comparison with the above baselines on MovieLens and Bookcrossing data sets.

### *5.3.1 Superiority over MF Methods*

We first evaluate our proposed two-level MF model comparing to the above MF series, with the results shown in Table 3, which clearly demonstrate our proposed TLMF outperforms other MF methods which only utilize user item rating matrix such as PMF, RSVD and ISMF. Specifically, TLMF respectively improves PMF by 1.7% and 5.89% on MovieLens and BookCrossing data sets with MAE metric and latent dimension as 50. With the same setting, the MAE performance of ISMF can be improved by 1.1% and 5.61% respectively on MovieLens and BookCrossing data. TLMF achieves better than RSVD as well, which is improved by 1.0% and 5.92%, respectively for MovieLens and BookCrossing data.

We then compare the proposed TLMF model with other three MF models EDMF, TFMF and TIMF which also consider item relations based on their textual information. From Table 3, we can clearly see that TLMF achieves better performance than the three models. In detail, TLMF respectively improves EDMF by 1.5% and 7% regarding MAE evaluation metric on MovieLens and BookCrossing data sets when latent dimension is 50. In the same setting, TFMF can also be improved by 0.9%, 6.44%, as well for TIMF by 0.7%, 2.34%.

Table 3: MF Comparisons on MovieLens and BookCrossing

| Data Set | Dim | Metrics | PMF | ISMF | RSVD | EDMF | TFMF | TIMF | TLMF |
|---|---|---|---|---|---|---|---|---|---|
| MovieLens | 50D | MAE | 0.685 | 0.679 | 0.678 | 0.683 | 0.677 | 0.675 | **0.668** |
| | | RMSE | 0.865 | 0.859 | 0.862 | 0.863 | 0.861 | 0.864 | **0.853** |
| | 10D | MAE | 0.688 | 0.685 | 0.683 | 0.697 | 0.689 | 0.684 | **0.676** |
| | | RMSE | 0.871 | 0.864 | 0.867 | 0.869 | 0.863 | 0.862 | **0.859** |
| Bookcrossing | 50D | MAE | 1.5128 | 1.5100 | 1.5131 | 1.5239 | 1.5183 | 1.4773 | **1.4539** |
| | | RMSE | 3.7452 | 3.7415 | 3.7648 | 3.7894 | 3.7850 | 3.7426 | **3.7262** |
| | 10D | MAE | 1.5135 | 1.5107 | 1.5134 | 1.5164 | 1.5128 | 1.5085 | **1.4836** |
| | | RMSE | 3.7483 | 3.7440 | 3.7659 | 3.7591 | 3.7483 | 3.7426 | **3.7425** |

Table 4: CF Comparisons on MovieLens and BookCrossing

| Data Set | Metrics | UBCF (Improve) | IBCF (Improve) | TLMF |
|---|---|---|---|---|
| MovieLens | MAE | 0.9027 (23.47%) | 0.9220 (25.4%) | **0.668** |
| | RMSE | 1.0022 (14.92%) | 1.1958 (34.28%) | **0.853** |
| Bookcrossing | MAE | 1.8064 (35.25%) | 1.7865 (33.26%) | **1.4539** |
| | RMSE | 3.9847 (25.82%) | 3.9283 (20.21%) | **3.7262** |

### *5.3.2 Superiority over CF Methods*

In addition to the MF methods, we also compare our proposed TLMF model with two different CF methods UBCF and IBCF. In this experiment, we fix the latent dimension to 50 for TLMF model. On MovieLens, the results in Table 4 indicate that TLMF can respectively improve 23.47% and 25.4% regarding MAE, and 14.92% and 34.28% in terms of RMSE. Similarly compared to UBCF and IBCF, on Bookcrossing data set, the results show that the TLMF can reach improvements respectively 35.25% and 33.26% regarding MAE, and 25.82% and 20.21% regarding RMSE. Therefore, this experiment clearly demonstrates that our proposed TLMF performs better than UBCF and IBCF methods on both data sets. The improvements are contributed by the consideration of item semantic relations in RS.

### *5.3.3 Impact of Parameter $\alpha$*

Parameter $\alpha$ controls the influence of semantic relations between items in TLMF model. A bigger value of $\alpha$ in the objective function of Eqn. 12 indicates a higher impact of the items' relations. To select the optimum parameter $\alpha$, we depict the MAE and RMSE changing trends of TLMF model when $\alpha$ ranges in [0,1]. Fig. 3 shows the impacts of parameter $\alpha$ with latent dimension $d$=50 on MovieLens and BookCrossing data sets. Experimental results show that the proper values of $\alpha$ for MovieLens and BookCrossing are respectively 0.3 and 0.6.

### *5.3.4 Impact of Dimension of Latent Vectors*

Parameters $d$ and $K$ control the dimension of latent vectors respectively for the upper and lower level MF. In this paper, we also investigate the impact of the dimension

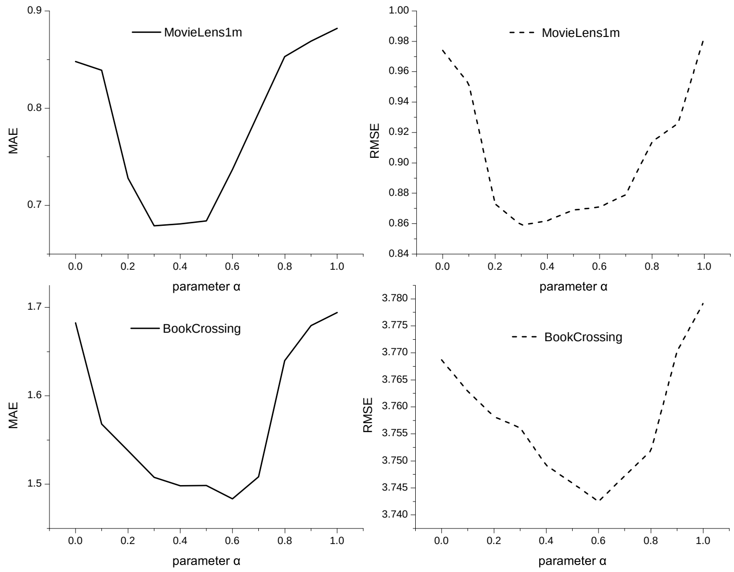


Fig. 3: Impact of Parameter $\alpha$ on MovieLens and BookCrossing

parameters by fixing parameter $\alpha$ to 0.3 and 0.6 for MovieLens and BookCrossing. Fig. 4 clearly shows the impact of parameter $d$, indicating that the performance regarding MAE and RMSE would be decreased after the parameter $d$ reaching 50 and 60 for MovieLens and BookCrossing, respectively. Similarly, Fig. 5 shows the MAE and RMSE changing trends for different parameter $K$ of lower level MF with $w_m$=0.01, indicating that optimum values are 175 and 150 respectively for MovieLens and BookCrossing data. It is also noticed that the optimum parameter value of $K$ is bigger than parameter $d$, which is possibly caused by big textual corpus having much more implicit features to depict.

### *5.3.5 Impact of Neighborhood Size*

In addition to the above parameters, the neighborhood size of items also influence the optimization process of the objective function. Fig. 6 shows the effect of the neighborhood size of items for MovieLens and BookCrossing data sets. The experimental results indicate that the MAE and RMSE decrease sharply with the increase of the neighborhood size of items until reaching a steady point. Continually increasing the neighborhood size would not improve the performance after the steady point. From the experiments, we can see that the best neighborhood size of items for MovieLens and BookCrossing is respectively 60 and 300.

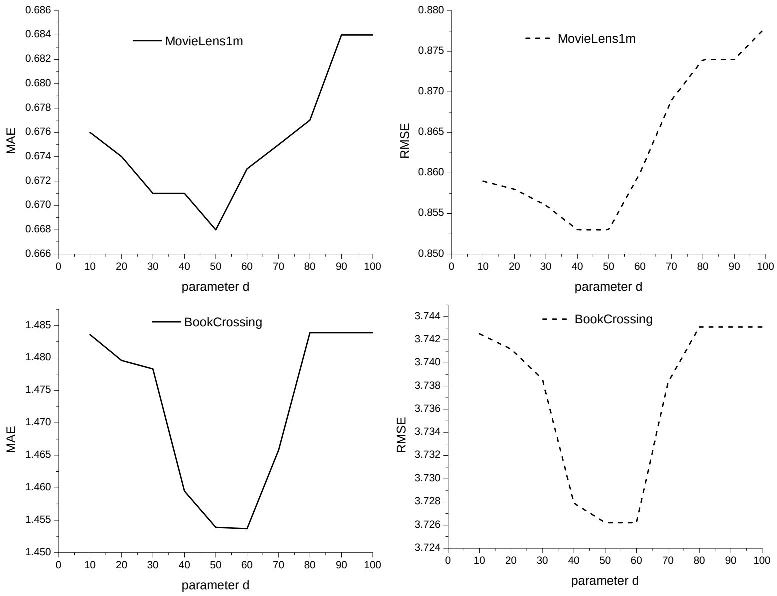


Fig. 4: Impact of Parameter $d$ on MovieLens and BookCrossing

Table 5: MF Comparisons on Cold Start Items

| Data Set | Metrics | PMF | ISMF | RSVD | EDMF | TFMF | TIMF | TLMF |
|---|---|---|---|---|---|---|---|---|
| MovieLens | MAE | 0.697 | 0.693 | 0.692 | 0.694 | 0.687 | 0.689 | **0.671** |
| | RMSE | 0.878 | 0.874 | 0.876 | 0.878 | 0.875 | 0.872 | **0.868** |
| Bookcrossing | MAE | 1.5332 | 1.5318 | 1.5338 | 1.5445 | 1.5385 | 1.4978 | **1.4751** |
| | RMSE | 3.7655 | 3.7618 | 3.7852 | 3.8098 | 3.8055 | 3.7632 | **3.7470** |

### *5.3.6 Cold Start Recommendation*

Besides the above experiments, we also compared the effectiveness of solving the problem of cold start items. To select the cold start items, we filter the data sets by selecting the items which are rated less than 50 times as cold start items. To evaluate the effectiveness of our proposed approach on cold start items, we compare the RMSE and MAE results with other benchmark methods as showed in Table 5. The parameters are respectively fixed as $\alpha$=0.3, $d$=50, $K$=175, $\mathbb{N}$=60 for MovieLens data, and $\alpha$=0.6, $d$=60, $K$=150, $\mathbb{N}$=300 for BookCrossing data. The comparison results clearly demonstrate that the proposed TLMF method outperforms other benchmark methods, which is resulted from considering the textual semantic relations between items.

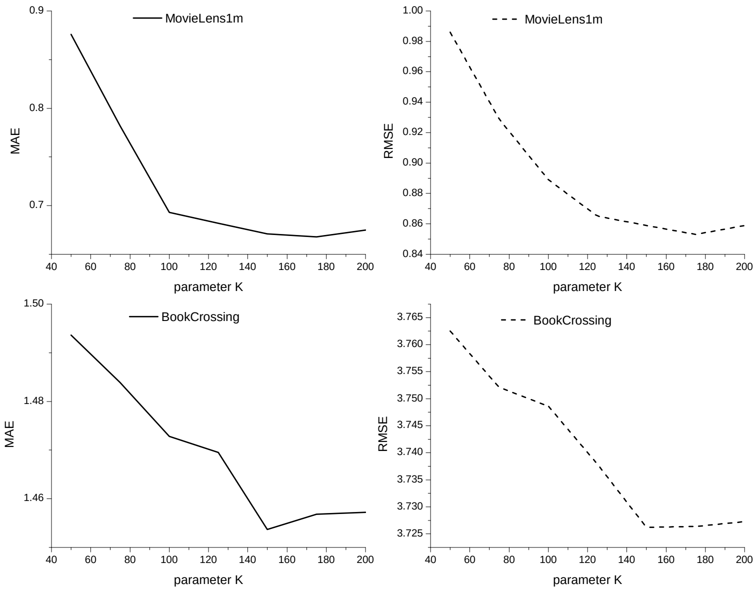


Fig. 5: Impact of Parameter $K$ on MovieLens and BookCrossing

### *5.3.7 Discussion*

From the above experiments, we demonstrate the effectiveness of our proposed TLMF model and the superiority over MF and CF methods. Generally, we can conclude that TLMF is more effective than the benchmark MF and CF approaches regarding MAE and RMSE for different latent dimensions, due to the strength of semantic relations between items. Besides, the comparisons with EDMF, TFMF and TIMF show that the consideration of missing words in lower level MF is beneficial for computing item semantic relations and further helpful for improving recommendation qualities.

## 6 Conclusion

In this paper, we first studied semantic relations between items based on textual information to improve recommendation quality. Actually, the semantic relations were captured by modelling the term document matrix with TF*IDF values by textual MF approach which considered missing words to enhance the capability of semantic analysis. A two-level matrix factorization model was then proposed to incorporate the semantic relations between items and the rating matrix. The proposed two-level MF model, on one hand, took the advantage of latent semantic analysis of textual MF. On the other hand, it also balanced the traditional rating matrix based MF model.

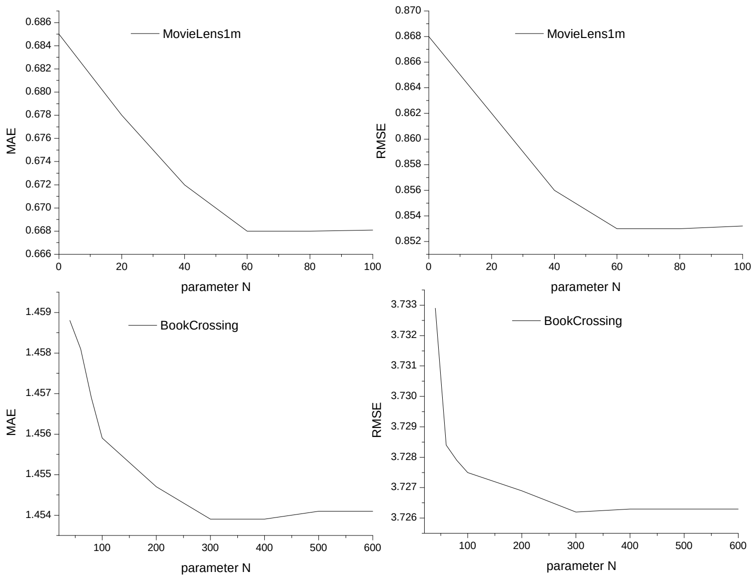


Fig. 6: Impact of Neighborhood Size of Items on MovieLens and BookCrossing

The experiments conducted on the real data sets demonstrated the superiority of the proposed TLMF model and suggested that semantic relations could be effectively applied in RS. To further enhance the performance of the proposed model, other deeper semantic computation methods for items such as ontologies can be explored in the future. To make the proposed model more beneficial for the recommendation community, it would be awesome to implement the TLMF model on the big data platform such as Apache Hadoop in future work.

**Acknowledgements** This work is sponsored in part by Australian Research Council Discovery Grants (DP1096218 and DP130102691) and ARC Linkage Grant (LP100200774). We appreciate the valuable comments from all the reviewers.

### Compliance with Ethical Standards and Disclosure of Conflicts of Interest

**Funding** This study was funded by Australian Research Council Discovery Grants (DP1096218 and DP130102691) and ARC Linkage Grant (LP100200774).

**Conflict of Interest** Author Fangfang Li has received research grants from University of Technology Sydney. Author Longbing Cao is the Chair of ACM SIGKDD Australia and New Zealand Chapter, and a Senior Member of IEEE, SMC Society and Computer Society. Author Guandong Xu is an assist. Editor-in-Chief of World Wide Web Journal.

**Research involving Human Participants and/or Animals** The authors declare that there are no involved human participants and animals.

**Informed consent** The authors declare that they consent to submit the manuscript to Neural Computing and Applications.

## References


1. Agarwal, D., Chen, B.C.: Regression-based latent factor models. In: KDD, pp. 19–28 (2009)
2. Agarwal, D., Chen, B.C.: flda: matrix factorization through latent dirichlet allocation. In: WSDM, pp. 91–100 (2010)
3. Alter, O., Brown, P.O., Botstein, D.: Singular value decomposition for genome-wide expression data processing and modeling. Proceedings of the National Academy of Sciences (PNAS) **97**(18) (2000)
4. Blei, D.M., Ng, A.Y., Jordan, M.I., Lafferty, J.: Latent dirichlet allocation. Journal of Machine Learning Research **3**, 2003 (2003)
5. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Trans. Inf. Syst. **22**(1), 143–177 (2004). DOI 10.1145/963770.963776. URL `http://doi.acm.org/10.1145/963770.963776`
6. Gantner, Z., Drumond, L., Freudenthaler, C., Rendle, S., Schmidt-Thieme, L.: Learning attribute-to-feature mappings for cold-start recommendations. In: ICDM, pp. 176–185 (2010)
7. Guo, W., Diab, M.: A simple unsupervised latent semantics based approach for sentence similarity. In: Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, SemEval '12, pp. 586–590. Association for Computational Linguistics, Stroudsburg, PA, USA (2012). URL `http://dl.acm.org/citation.cfm?id=2387636.2387735`
8. Guo, W., Diab, M.T.: Modeling sentences in the latent space. In: The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers, pp. 864–872. The Association for Computer Linguistics (2012). URL `http://www.aclweb.org/anthology/P12-1091`
9. Guo, W., Li, H., Ji, H., Diab, M.T.: Linking tweets to news: A framework to enrich short text data in social media. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers, pp. 239–249. The Association for Computer Linguistics (2013). URL `http://aclweb.org/anthology/P/P13/P13-1024.pdf`
10. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '99, pp. 50–57. ACM, New York, NY, USA (1999). DOI 10.1145/312624.312649. URL `http://doi.acm.org/10.1145/312624.312649`
11. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Y. Li, B. Liu, S. Sarawagi (eds.) KDD, pp. 426–434. ACM (2008)
12. Landauer, T.K., Foltz, P.W., Laham, D.: An Introduction to Latent Semantic Analysis. Discourse Processes (25), 259–284 (1998)
13. Li, F., Xu, G., Cao, L.: Coupled item-based matrix factorization. In: WISE (1), Lecture Notes in Computer Science, pp. 1–14. Springer (2014)
14. Li, F., Xu, G., Cao, L., Fan, X., Niu, Z.: Cgmf: Coupled group-based matrix factorization for recommender system. In: X. Lin, Y. Manolopoulos, D. Srivastava, G. Huang (eds.) WISE (1), *Lecture Notes in Computer Science*, vol. 8180, pp. 189–198. Springer (2013)
15. Ma, H.: An experimental study on implicit social recommendation. In: SIGIR, pp. 73–82 (2013)
16. Ma, H., Yang, H., Lyu, M.R., King, I.: Sorec: social recommendation using probabilistic matrix factorization. In: J.G. Shanahan, S. Amer-Yahia, I. Manolescu, Y. Zhang, D.A. Evans, A. Kolcz, K.S. Choi, A. Chowdhury (eds.) CIKM, pp. 931–940. ACM (2008)
17. Melville, P., Sindhwani, V.: Recommender systems. In: Encyclopedia of Machine Learning, pp. 829–838 (2010)
18. Menon, A.K., Elkan, C.: A log-linear model with latent features for dyadic prediction. In: ICDM, pp. 364–373 (2010)

19. Nati, N.S., Jaakkola, T.: Weighted low-rank approximations. In: In 20th International Conference on Machine Learning, pp. 720–727. AAAI Press (2003)
20. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: THE ADAPTIVE WEB: METHODS AND STRATEGIES OF WEB PERSONALIZATION. VOLUME 4321 OF LECTURE NOTES IN COMPUTER SCIENCE, pp. 325–341. Springer-Verlag (2007)
21. Rendle, S.: Factorization machines. In: ICDM, pp. 995–1000 (2010)
22. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: Advances in Neural Information Processing Systems, vol. 20 (2008)
23. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Adv. Artificial Intellegence **2009** (2009)
24. Wang, C., Blei, D.M.: Collaborative topic modeling for recommending scientific articles. In: KDD, pp. 448–456 (2011)
25. Wang, C., Cao, L., Wang, M., Li, J., Wei, W., Ou, Y.: Coupled nominal similarity in unsupervised learning. In: CIKM, pp. 973–978 (2011)
26. Wang, C., She, Z., Cao, L.: Coupled attribute analysis on numerical data. In: IJCAI (2013)
27. Wang, C., She, Z., Cao, L.: Coupled clustering ensemble: Incorporating coupling relationships both between base clusterings and objects. In: ICDE, pp. 374–385 (2013)
28. Wu, H.C., Luk, R.W.P., Wong, K.F., Kwok, K.L.: Interpreting tf-idf term weights as making relevance decisions. ACM Trans. Inf. Syst. **26**(3), 13:1–13:37 (2008). DOI 10.1145/1361684.1361686. URL `http://doi.acm.org/10.1145/1361684.1361686`
29. Wu, L., Chen, E., Liu, Q., Xu, L., Bao, T., Zhang, L.: Leveraging tagging for neighborhood-aware probabilistic matrix factorization. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12, pp. 1854–1858. ACM, New York, NY, USA (2012). DOI 10.1145/2396761.2398531. URL `http://doi.acm.org/10.1145/2396761.2398531`
30. Yang, X., Steck, H., Liu, Y.: Circle-based recommendation in online social networks. In: KDD, pp. 1267–1275 (2012)
31. Yu, Y., Wang, C., Gao, Y., Cao, L., Chen, Q.: A coupled clustering approach for items recommendation. In: PAKDD (2), pp. 365–376 (2013)