An Extended-Tag-Induced Matrix Factorization Technique for Recommender Systems

Huirui Han [1,2], Mengxing Huang [1,2,*], Yu Zhang [1,2] and Uzair Aslam Bhatti[1,2]

[1] State Key Laboratory of Marine Resource Utilization in South China Sea, Hainan University, Haikou 570228, China; hanhr26@163.com (H.H.); yuzhang_nwpu@163.com; (Y.Z.) uzairaslambhatti@hotmail.com (U.Z.B)

[2] College of Information Science & Technology, Hainan University, Haikou 570228, China

[*] Correspondence: huangmx09@163.com

Abstract: Social tag information has been used by recommender systems to handle the problem of data sparsity. Recently, the relationships between users/items and tags are considered by most tag-induced recommendation methods. However, sparse tag information is challenging to most existing methods. In this paper, we propose an Extended-Tag-Induced Matrix Factorization technique for recommender systems, which exploits correlations among tags derived by co-occurrence of tags to improve the performance of recommender systems, even in the case of sparse tag information. The proposed method integrates coupled similarity between tags, which is calculated by the co-occurrences of tags in the same items, to extend each item's tags. Finally, item similarity based on extended tags is utilized as an item relationship regularization term to constrain the process of matrix factorization. MovieLens dataset and Book-Crossing dataset are adopted to evaluate the performance of the proposed algorithm. The results of experiments show that the proposed method can alleviate the impact of tag sparsity and improve the performance of recommender systems.

Keywords: recommender system; collaborative filtering; matrix factorization; social tag

1. Introduction

Collaborative filtering is one of the most popular recommendation techniques. It is divided into two categories: memory-based collaborative filtering and model-based collaborative filtering. Memory-based collaborative filtering [1-4], known as neighbor-based methods, using some similarity measures, discovers neighbors of active users and neighbors of target items. Once the neighbors are found, the ratings of neighbors are usually used to predict the rating of target item by memory-based algorithms. Model-based filtering makes use of user-item rating matrix to learn a predictive model by statistical and machine learning methods firstly, thereafter the model predicts the rating of the target item. Recently, matrix factorization, which discovers latent preferences of users and items and deals with very large user-item rating matrix effectively, has developed as a very popular recommendation algorithm.

In the last years, developed recommendation systems use of different sources of information, including social information, social behaviors of users, information of items, etc., to provide recommendations of items to users [5]. For example, the contextual description was combined with usage patterns to predict behaviors of users and provide effective recommendation services [6]. The detailed categories of extra information integrated by hybrid recommendation methods are listed in Table 1.

Table 1. The categories of extra information

Categories	Detailed description
Social information	the "credibility" of users [7], social relationships of users discovered by social networks [8]
Social behaviors of users	Users' browsing behaviors [9], users' point of interest [10]
Opinions of users	Comments given by users [11,12]
Information of items	Items' reputations, semantic contents [6] and items' attributes [5,13]
Tag information	Tags annotated by users and tags provided by systems [14]

Beside the basic descriptions of users and items, tag information, which has been incorporated into hybrid CBF/CF algorithms by being used to calculate user-based and item-based similarity measures [14], is a kind of useful semantic information for recommendation systems.

Nowadays, collaborative tagging system, which improves the interaction between users and systems, has been applied in several recommender systems. In collaborative tagging system, users are allowed to annotate sources with some specific tags in terms of their comprehensions. To some extent, characteristics of sources can be reflected by tags as users utilize tags to annotate the features and categories of sources. Therefore, more accurate classification of sources can be achieved through analyzing tag information. Meanwhile, personalized preferences of users can be explored by analyzing tag information, because the diversity among annotating tags represents the different personalized information among users. Hence, tag information, as a kind of vital data, brings new challenges and opportunities to recommender systems.

At present, many studies [15] focus on utilizing tags to improve recommendation algorithms and provide more personalized recommendations. To relieve the influence of rating data sparsity, tag information has been introduced to reinforce the relationship of users and items. Peng et al. [16] proposed the method of probabilistic model by considering each tag as a specific topic and measuring the probability of a tag used by a user to annotate an item. Blaze et al. [17] proposed a tensor factorization exploiting the content of items and users' tag assignments through a relevance feedback mechanism for identifying the optimal number of conceptually similar items. Tag-aware method [18] regarded tagging information as a data source for extending user-item rating vectors. TagiCoFi proposed by Yi Zhen et al. [19] exploits tagging information to regularize the matrix factorization procedure of probabilistic matrix factorization. Huang et al. [20] proposed a content-based collaborative filtering using tagging information to alleviate cold start problem and sparsity problem in the collaborative filtering recommendation algorithms. To provide an enhanced recommendation quality derived from user-created tags, Kim et al. [21] proposed a collaborative filtering method which uses collaborative tags as an approach to grasp and filter users' preferences for items. Nguyen et al. [22] studied different content-boosted matrix factorization techniques which integrate content information into the matrix factorization collaborative filtering methods. They also found that these approaches not only improve recommendation accuracy, but also provide useful insights about the contents, as well as make the recommendation more easily interpretable. Huang et al. [23] constructed a personalized user interest by incorporating frequency, recency and tag-based information and performed collaborative recommendations using user's social network in social resource sharing websites. Rawashdeh et al. [24] showed a novel personalized search algorithm building two models of which one is user-tag relation model that reflects how a certain user assign tags which are similar to a given tag, the other one is tag-item relation model that captures how a certain tag is annotated to items which are similar to a given item. Kim et al. [25] proposed a recommender system based on graph model providing recommendations to a group of users instead of a single user, and this system not only considers positive feedbacks from users but also negative feedbacks from users. Three-factor factorization model is used to learn user preference vectors based on the tags annotated by users and item feature vectors based on the keywords corresponding to items [26]. The co-occurrences between tags annotated by users and keywords corresponding to items are utilized to create the relationships of tags and keywords, which requires all user have tags and all items have keywords.

Most existing methods compute similarities of users and similarities of items based on match of

tags, while few methods consider the tag sparsity problem. Tag sparsity is one of the difficult problems in collaborative recommendation algorithms based on tags. There are two causes of tag sparsity, one is user rarely annotate tags to items or merely annotate very few tags to items, and the other is some tags are different but similar in semantic due to different comprehensions of users. Because of the significant impact and applying value of tags to recommendation techniques, resolving the tag sparsity problem is a crucial task to improve recommendation techniques. As shown in Figure 1, User1 gives a rating to Item1 and annotate Item1 with "psychology" and "clever"; User2 gives a rating to Item2 and annotate Item2 with "clever" and "genius"; Item3 is merely annotated with "genius" by User1 and without any rating. At that time, it is difficult to obtain the similarities of these three items based on ratings and match of tags.
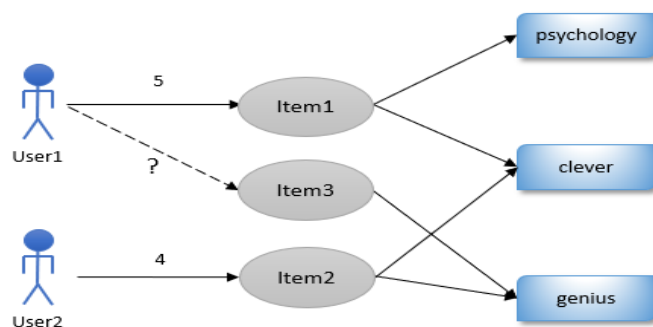

Figure 1. The example of the recommendation with the association of tags.

Fang et al. [27] considered correlations among tags to construct a tag co-occurrence matrix transfer model to regularize the procedure of learning user latent factors and item latent factors. However, the similarities of items based on similar tags is not considered, which is not accurate to capture the item latent factors. Therefore, to alleviate the deviation of similarities of items based on sparse tags, an Extended-Tag-Induced Matrix Factorization (ETIMF) method, which extends tags of items and exploits extended tags to generate similarities of items, is proposed in this paper. Considering the latent feature factors between coupling items similar when their corresponding extended tags are similar, the proposed method uses extended tags to constrain the process of matrix factorization via an item relationship regularization term. The recommender system using the proposed method contain two processing modules, namely tags processing module and recommendation module. Tags processing module carries out the inputting tags and extends them. Once recommendation module receives the extended tags, it uses them to control the process of rating matrix factorization. After that, it predicts and ranks the ratings of the target user. Finally, the Top N items would recommend to the target user. The overview of the recommender system is shown in Figure 2. The results of experiments show that the proposed method can alleviate the impact of tag sparsity and improve the performance of recommender systems.
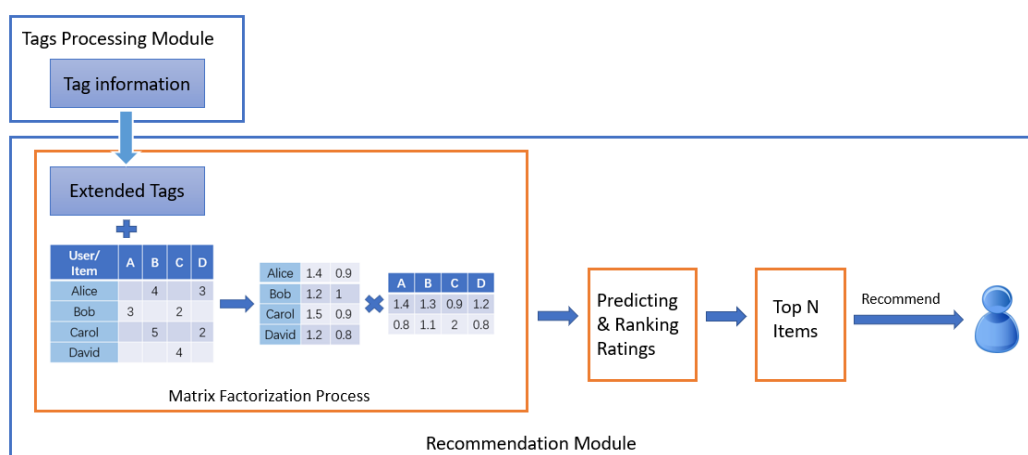

Figure 2. The overview of the recommender system with the association of tags.

2. Materials and Methods

2.1. Primary Definition

In a typical scenario, a recommender system contains a set of M user $U = \{u_1, u_2, \cdots, u_m\}$ and a set of N items $I = \{v_1, v_2, \cdots, v_n\}$. The user preference of items is usually represented by a user-item rating matrix $R \in \mathbb{R}^{m \times n}$. Each entry $R_{u,v}$ denotes the rating given by user u on item v. In general, $R_{u,v}$ is an integer and falls into $1,5$, where $R_{u,v} = 0$ represents user u has not yet given rating to item i. Higher values of $R_{u,v}$ means user u has a better satisfaction on item v.

The tags of an item are denoted by a set of tags $T_I = \{t_1, t_2, \cdots, t_k\}$ and the sizes of the sets on various items can be different. Table 2 shows the tags of some items.

Table 2. The tags of some items.

Items	Tags
I₁	{ history, space, genius, psychology }
I₂	{ revenge, violent, visual, psychology}
I₃	{ teen, adventure, dancing, fantasy }
I₄	{ romance, touching, fantasy}
I₅	{ robots, sci-fi, quirky, genius}

2.2. Matrix Factorization

Matrix factorization is widely used and applied to research due to its high efficiency in resolving large-scale user-item rating matrices. Based on the assumption that latent preferences of users and latent characteristics of items can be represented by a certain number of factors, matrix factorization algorithm decomposes user-item rating matrix into two low rank latent feature matrices, namely $P \in \mathbb{R}^{k \times m}$ and $Q \in \mathbb{R}^{k \times n}$, where $k \ll \min(m, n)$, and then uses P and Q to rebuild a predictive rating matrix $R^* \in R^{m \times n}$. As a result,

$$R^* = P^T Q = \begin{bmatrix} p_1^T \\ p_2^T \\ \cdots \\ P_m^T \end{bmatrix} [q_1, q_2, \cdots, q_n] \tag{1}$$

where the column vectors p_u and q_v represents the K-dimensional user latent feature vector corresponding to user u and K-dimensional item latent feature vector corresponding to item v respectively. Once getting the low rank vectors, we can use the inner product of p_u and q_v to estimate the rating given by user u to item v, which is $R_{u,v}^* = p_u^T q_v$.

The user latent feature matrix and the item latent feature matrix are learning by minimizing the following loss function,

$$\ell = \sum_{(u,v) \in \Omega} (R_{u,v} - p_u^T q_v)^2 + \lambda_1 \| P \|_F^2 + \lambda_2 \| Q \|_F^2 \tag{2}$$

where λ_1 and λ_2 denotes regularization parameters [28], which is used to prevent overfitting. $\| \cdot \|_F^2$ is Frobenius norm [29], and Ω indicates the set of the (u, v) tuples for known ratings.

Generally, the stochastic gradient descent algorithm (SGD) [30] is applied to seek a local minimum of loss function given by Equation (2). The matrix factorization is a widely applied collaborative filtering method, thus the matrix factorization algorithm is regard as baseline approach.

However, data sparsity and cold start problem are main challenges of matrix factorization recommendation algorithm.

2.3. Extended-Tag-Induced Matrix Factorization

Extended-Tag-Induced Matrix Factorization method (ETIMF) aiming at recommending items has three stages. Firstly, tag-tag co-occurence matrix is created by analyzing co-occurrences of tags in the same items. Secondly, extended tag vectors of each coupled items are rebuilt according to tag-tag co-ocurrence matrix and item similarity is estimated by the extended tag vectors. Finally, the item similarity obtained by the second stage is used as a regularization term to constrain the process of matrix factorization.

2.3.1. Tag-Tag Co-Occurrence Matrix

Tags of an item assigned by different users might be different due to the various comprehensions of users. Thus, the similarities of users and the similarities of items only measured by match of users' tags and items' tags are inaccurate. Each two tags' frequencies of co-occurrence in items can be used to evaluate similarity of the two tags. Therefore, tag-tag co-occurrence matrix is created to extend tags. Cosine similarity is used to evaluate co-occurrence distribution of each two tags as follow:

$$p(t,z) = \frac{\sum_{i \in N(t) \cap N(z)} n_{t,i} \bullet n_{z,i}}{\sqrt{\sum_{i \in N(t)} n_{t,i}^2 \bullet \sum_{i \in N(z)} n_{z,i}^2}} \tag{3}$$

where $n_{t,i}$ indicates the number of tag t annotated to item i and $n_{z,i}$ indicates the number of tag z annotated to item i. $N(t)$ is the set of items annotated as tag t and $N(z)$ is the set of items annotated as tag z. $N(t) \cap N(z)$ is the set of items annotated as tag t and tag z. $p(t,z)$ falls into [0,1] and $p(t,z)$ closer to 1 represents tag t and tag z are more similar. The co-occurrence distribution of tag "sci-fi" and other tags is evaluated by Equation (3) and tags co-occurring with "sci-fi" whose co-occurrecne probability ranking top 10 are listed in Table 3. These 10 tags are relevant to "sci-fi" in semantic.

Definition 1: Tag-tag co-occurrence matrix $T \in \mathbb{R}^{k \times k}$:

$$T = \begin{bmatrix} p(t_1,t_1) & p(t_1,t_2) & \cdots & p(t_1,t_k) \\ p(t_2 t_1) & p(t_2,t_2) & \cdots & p(t_2,t_k) \\ \ldots & \ldots & \ldots & \ldots \\ p(t_k,t_1) & p(t_k,t_2) & \cdots & p(t_k,t_k) \end{bmatrix} \tag{4}$$

where each entry of above matrix $T_{ij} = p(t_i,t_j)$ represents the similarity of tag t_i and tag t_j. In terms of tag-tag co-occurrence matrix, the relationships of all tags can be obtained.

Table 3. TOP-10 tags co-occurring with "Sci-fi".

Rank	Tags	Co-Occurrence Probability
1	space	0.5917
2	space opera	0.5262
3	aliens	0.5136
4	special effects	0.5116
5	action	0.4965
6	future	0.4914
7	spaceships	0.4855
8	science fiction	0.4822
9	futuristic	0.4766
10	far future	0.4577

2.3.2. Tag Vectors of Item to Item

Before evaluating item similarities based on tags, tags of each two items are mapped to the shared tag space of these two items. We define vector $n_{i,j} = [n_{i,j}^{(1)}, \dots, n_{i,j}^{(N(T_i)+N(T_j)-N(T_{i\cap j}))}]$ as tag vector of item i corresponding to item j and vector $n_{j,i} = [n_{j,i}^{(1)}, \dots, n_{j,i}^{(N(T_i)+N(T_j)-N(T_{i\cap j}))}]$ as tag vector of item j corresponding to item i. Here, $N(T_i)$ and $N(T_j)$ are the number of tags of item i and the number of tags of item j respectively, and $N(T_{i\cap j})$ is the number of total tags annotated to item i and item j. The entry $n_{i,j}^{(k)}$ is the frequency of tag k annotated to item i and $n_{i,j}^{(k)} = 0$ if tag k has been annotated to item j but not been annotated to item i. Similarly, the entry $n_{j,i}^{(k)}$ is the frequency of tag k annotated to item j and $n_{j,i}^{(k)} = 0$ if tag k has been annotated to item i but not been annotated to item j.

2.3.3. Extended tag vectors of item to item

The item similarities are inaccurate based on above tag vectors of item to item due to the diversity of tags. Tag-tag co-occurrence matrix is used to extend tags and reconstruct extended tag vectors of item to item. For the tag z, which has been annotated to item j but has not item i, we estimate the possible frequency of tag z annotated to item i according to co-occurrence distributions between tag z and all tags annotated to item i. We estimate the possible frequency as follow:

$$n_{i,j}^{z*} = \frac{\sum_{t \in T_i}(n_i^t * p(t,z))}{N_{t \in T_i}} \tag{5}$$

where T_i denotes the set of tags annotated to item i and n_i^t denotes the frequency of tag t annotated to item i. $p(t,z)$ is the co-occurrence probability between tag t and tag z, namely the entry (t,z) of tag-tag co-occurrence matrix. $N_{t \in T_i}$ indicates the number of total tags annotated to item i.

2.3.4. Similarities of items based on extended tags

After recreating extended tag vectors of item to item, cosine similarity is used to evaluate the similarity of each two items as follow:

$$Sim(v_i, v_j) = \frac{\sum_{k \in T_{i,j}^*} n_{ik} \bullet n_{jk}}{\sqrt{\sum_{k \in T_i^*} n_{ik}^2} \bullet \sqrt{\sum_{k \in T_j^*} n_{jk}^2}} \tag{6}$$

where $T_{i,j}^*$ indicates the set of tags shared by item i and item j after extending tags. T_i^* and T_j^* indicate the set of tags annotated to item i and the set of tags annotated to item j, respectively. $sim(i,j)$ belongs to $[0,1]$, and item i is more similar to item j if the value of $sim(i,j)$ is closer to 1.

Definition 2: item-item similarity matrix based on tags $S \in \mathbb{R}^{N \times N}$:

$$S = \begin{bmatrix} sim(v_1,v_1) & sim(v_1,v_2) & \cdots & sim(v_1,v_n) \\ sim(v_2,v_1) & sim(v_2,v_2) & \cdots & sim(v_2,v_n) \\ \dots & \dots & \dots & \dots \\ sim(v_n,v_1) & sim(v_n,v_2) & \cdots & sim(v_n,v_n) \end{bmatrix} \tag{7}$$

where the entry $S_{ij} = sim(v_i, v_j)$ of this matrix indicates the similarity of item i and item j.

2.3.5. The process of matrix factorization

The main idea of the proposed algorithm is to utilize similarities of items based on extended tags to regularize the process of matrix factorization, and consequently deal with tag sparsity problem and cold start item problem. The similarities of items based on extended tags are converted to an item

relationship term. We suppose that two item latent feature vectors q_i and q_j are similar if these two items have similar tags according to extended tags.

To make two item latent vectors q_i and q_j as similar as possible if they are annotated as similar tags according to extended tags, an item relationship regularization term based on extended tags is added into matrix factorization to constrain the baseline matrix factorization (MF).

Definition 3: The item relationship regularization term:

$$\frac{\beta}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}S_{i,j}\parallel q_i - q_j\parallel_F^2 \tag{8}$$

where β indicates a regularization parameter to control the influence of similarities of items based on extended tags. $S_{i,j}$ indicates the similarity between item i and item j based on extended tags. Higher values of $S_{i,j}$ illustrate the distance between two item latent feature vectors is relatively shorter and vice versa. Therefore, the item regularization term makes two latent feature vectors closer if these two items have more similar tags.

Then, we convert the item relationship regularization term as follow:

$$\begin{aligned}
&\frac{\beta}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}S_{i,j}\parallel q_i - q_j\parallel_F^2\\
&=\frac{\beta}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}[S_{i,j}\sum_{k=1}^{K}(q_{ki}-q_{kj})^2]\\
&=\frac{\beta}{2}\sum_{k=1}^{K}Q_{k*}LQ_{k*}^T\\
&=\frac{\beta}{2}tr(QLQ^T)
\end{aligned} \tag{9}$$

where $Q = [q_1, q_2, ..., q_n]$ is item latent feature matrix which is consist of n item latent feature vectors. The dimensions of each item latent feature vector are K, thus q_{ki} is the value of latent feature vector of item i in the *k*th demension. Similarly q_{kj} is the value of latent feature vector of item j in the k th dimension. $L = D - S$ represents the Laplacian matrix [30] and D is a diagonal matrix with diagonal elements $D_{ii} = \sum_j S_{ij}$. $tr(\bullet)$ is the function to calculate the trace of inputting matrix.

After adding item relationship regularization term into loss (Equation (2)), the proposed extended-tag-induced matrix factorization method can be formulated as follow:

$$\ell_1 = \min_{P,Q}\frac{1}{2}\sum_{(u,i)\in\Omega}(R_{u,i}-p_u^T q_i)^2 + \frac{\lambda_1}{2}\parallel P\parallel_F^2 + \frac{\lambda_2}{2}\parallel Q\parallel_F^2 + \frac{\beta}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}S_{i,j}\parallel q_i - q_j\parallel_F^2 \tag{10}$$

where λ_1 and λ_2 is the regularization parameter which controls complexity and prevents overfitting [18]. Combining Equations (9) and (10), the loss function is converted to

$$\ell_2 = \min_{P,Q}\frac{1}{2}\parallel W\circ(R-P^T Q)\parallel_F^2 + \frac{\lambda_1}{2}\parallel P\parallel_F^2 + \frac{\lambda_2}{2}\parallel Q\parallel_F^2 + \frac{\beta}{2}tr(QLQ^T) \tag{11}$$

where $\circ$ is the Hadamard product, W is the indicator matrix and $W_{u,i}=1$ indicates user u has given a rating to item i.

Through gradient descent algorithm [31], the local minimum of loss function can be obtained. In each iteration, the gradients of P and Q are shown by Equations (12) and (13), respectively.

$$\frac{\partial\ell_2}{\partial P} = -Q(W\circ W\circ R)^T + Q(W\circ W\circ(P^T Q))^T + \lambda_1 P \tag{12}$$

$$\frac{\partial\ell_2}{\partial Q} = -P(W\circ W\circ R) + P(W\circ W\circ(P^T Q)) + \lambda_2 Q + \beta QL \tag{13}$$

To summarize, the proposed extended-tag-induced matrix factorization algorithm is described in Algorithm 1.

Algorithm 1. The Framework of Proposed Recommendation Algorithm ETIMF

Input: User-item rating matrix $R \in \mathbb{R}^{m \times n}$, the set of tags $T_I = \{t_1, t_2, \cdots, t_k\}$, the dimension of

latent feature vector K, the number of iteration W, the step size of gradient descent η,

parameters λ_1, λ_2 and β.

Output: The user latent feature matrix P and the item latent feature matrix Q.

1: Compute tag-tag similarity matrix by using Equation (3)

2: Create tag vectors of item to item

3: Recreate extended tag vectors of item to item

4: Using Equation (6) compute similarities of items according to extended tag vectors of

item to item

5: Initialize $P \in \mathbb{R}^{k \times m}$ and $Q \in \mathbb{R}^{k \times n}$ randomly

6: for w = 1 to W do

7:　　for k = 1 to K do

8:　　　　$P_{*k}^{w} \leftarrow P_{*k}^{w-1} - \eta \dfrac{\partial \ell_2}{\partial P_{*k}^{w-1}}$

9:　　　　$Q_{k*}^{w} \leftarrow Q_{k*}^{w-1} - \eta \dfrac{\partial \ell_2}{\partial Q_{k*}^{w-1}}$

10:　　end for

11: end for

---

3. Results

MovieLens and Book-Crossing datasets were utilized to perform several groups of experiments. The first group consisted of comparing experiments for evaluating the performance of the proposed method. The second group analyzed the influences of regularization parameter β and the dimension of latent feature vector K. The third group was performed to discuss efficiency performances of different methods. The proposed method in the case of sparse tags was evaluated by the fourth group. The last group was designed to verify whether the proposed method can alleviate the cold start item problem.

3.1. Dataset

Several datasets have been adopted to evaluate the performance of recommendation algorithms. MovieLens 20M dataset [32] and Book-Crossing datasets [33] were applied to perform the experiments. MovieLens 20M dataset, which contains 20 million ratings and 465,000 tag applications applied to 27,000 movies by 138,000 users, is provided by GroupLens in 2015. Book-Crossing dataset contains 278,858 users providing 1,149,780 ratings about 271,379 books.

First, we filtered the original datasets to gain appropriate experimental datasets. For MovieLens 20M dataset, the movies rated by at least 20 users were extracted as usual movies. Tags annotated by at least five users and assigned to at least five usual movies were kept as distinct tags. For users, we only kept those users who annotated at least three distinct tags in the tagging history as distinct users. For movies, we only kept those movies that were annotated by distinct users and annotated as distinct tags as final distinct movies. For Book-Crossing dataset, the books rated by at least 20 users were extracted from the original dataset as usual books. For tag information, authors who had written at least three usual books and presses that hadpublished at least 15 usual books were kept as distinct tags. We only considered those items with tags in the experiments, but ETIMF method can be used in the case that items are without any tags. For items without any tags, the similarities between them and other items are 0, which means the item regularization term (Equation (9)) will have no effect on

those items. Consequently, ETIMF method changes into basic MF. For evaluating the effectiveness of the proposed method, we only kept those items which had been annotated as tags and given ratings from original datasets.

After filtering the datasets, for MovieLens 20M dataset, 2,161 distinct movies were kept, of which 999 were selected as the final distinct movies; for Book-Crossing dataset, 122,433 distinct books were kept, of which 10000 were selected as the final distinct books. Each movie is rated using a discrete scale from 1 to 5 in MovieLens dataset, and each book is rated using a discrete scale from 1 to 10 in Book-Crossing dataset. General statistics about the final datasets are summarized in Table 4.

In the experiments, the above datasets were required to be divided into training set which is for learning the parameters of models and test set which is for evaluating the models. Thus, the rating records were randomly split into two parts, each of which contained 50% of known ratings. One part was used as a test set, which was kept the same in all experiments. The other part was used as data pool to generate different training sets. For example, the size of the training set of 20% represents 20% of records were selected from the data pool as a training set. For each training set size, 10 different training sets were selected from data pool to perform 10 experiments and the average result was the final result.

Table 4. Statistics of MovieLens 20M and Book-Crossing.

Statistic	MovieLens 20M	Book-Crossing
No. of Ratings	375,873	12,931
No. of Users, m	7,711	1,851
No. of Items, n	999	10,000
No. of tag records	187,100	145,707
No. of tags	1968	1210

3.2. Performance Evaluation

Mean Absolute Error (MAE), which gives average absolute deviation between the real ratings and predictive ratings, was utilized to measure the recommendation quality of the proposed method compared with other recommendation algorithm. Formally,

$$MAE = \frac{\sum_i \sum_j W_{ij} \circ |R_{ij} - R_{ij}^*|}{\sum_i \sum_j W_{ij}}$$

where R_{ij} and R_{ij}^* represent real ratings and predictive ratings respectively. In general, a lower MAE means a lower deviation between real values and predictive values, namely ahigher quality of the recommendation algorithm.

3.3. Recommendation quality comparisons

To evaluate the performance of the proposed method, the following state-of-the-art approaches were chosen for comparison.

1. MF: Proposed by Koren et.al [34], MF learns user latent feature matrix and item latent feature matrix by minimizing the sum-of-squared errors between real ratings and predictive ratings. The number of latent feature vector K is 10.

2. TagiCoFi: Proposed by Yi Zhen et.al [19], TagiCoFi exploits tagging information to regularize the MF procedure of PMF. More specifically, it seeks to make two user latent feature vectors as similar as possible if the two users have similar tagging history. Here, we modify this algorithm to constrain item latent feature vectors by tagging information of items. The number of latent feature vector K is 10.

3. Tag Matrix Transfer(TMT) model: Proposed by Fang et.al [27], TMT model uses a tag co-occurrence matrix, which is generated by Bayesian method, as the third factor into matrix

factorization to improve the quality of recommender system. The number of latent feature vector K is the number of tags.

To make a fair comparison, we set parameters of each method according to respective references or based on the best performance of our experiments. For MovieLen 20M dataset, we set $\lambda_1 = \lambda_2 = 0.1$ and the learning rate η to be 0.05, and the control parameters of β in TagiCoFi and ETIMF were set to 0.1 and 1.3 respectively. For Book-Crossing dataset, we set $\lambda_1 = \lambda_2 = 0.1$ and the learning rate η to be 0.01, and the control parameters of β in TagiCoFi and ETIMF were set to be 0.1 and 4.5, respectively. In the comparisons, we used training sets size of 20%, 50% and 80% and report the average results on test sets.

The results of comparisons on these methods are shown in Table 5 and Table 6. We can observe that, no matter in which dataset, ETIMF achieves the best performance among all compared methods. Comparing with MF, the main difference of TagiCoFi, TMT and ETIMF lies in the extra tagging information. The results demonstrate that the recommendation quality can be improved by exploiting tagging information. Furthermore, ETIMF extends the tags of each item for supplementing the tagging information. The observations of the comparisons in two datasets demonstrate that ETIMF, which exploits extended tagging information, can further improve the recommendation quality. Therefore, extended tagging information is helpful to generate a better recommendation.

Table 5. The comparison of different recommendation method in MovieLens 20M.

Training set	K=10			K=30			K=50			K=number of tags
	MF	TagiCoFi	ETIMF	MF	TagiCoFi	ETIMF	MF	TagiCoFi	ETIMF	TMT
20%	0.7535	0.7480	0.7269	0.7487	0.7424	**0.7236**	0.7457	0.7397	**0.7231**	0.7242
50%	0.7380	0.7345	0.7231	0.7322	0.7289	**0.7220**	0.7322	0.7299	**0.7195**	0.7251
80%	0.7312	0.7280	0.7215	0.7239	0.7235	**0.7189**	0.7255	0.7234	**0.7166**	0.7198

Table 6. The comparison of different recommendation method in Book-Crossing.

Training set	K=10			K=30			K=50			K=number of tags
	MF	TagiCoFi	ETIMF	MF	TagiCoFi	ETIMF	MF	TagiCoFi	ETIMF	TMT
20%	2.1079	1.9175	1.7743	2.0600	1.9095	**1.7636**	2.0231	1.9025	**1.7445**	1.7710
50%	1.9741	1.8708	1.7315	1.9605	1.8508	**1.7261**	1.9029	1.8503	**1.7118**	1.7301
80%	1.9249	1.8375	1.6904	1.8717	1.8038	**1.6884**	1.8556	1.7903	**1.6847**	1.6895

3.4. The influence of tagging information

The value of the parameter β in ETIMF controls the influence of extended tags of items in learning each item latent feature vector. The higher value of β representswe put a larger weight in extended tags of items. The extended tags of items are added to learning item latent feature vectors and make two item latent feature vectors as similar as possible if the two item have similar tags and extended tags. When $\beta = 0$, ETIMF is reduced to basic matrix factorization. Hence, it is necessary to analyze the influence of β. In this section, we only use the training set of size 20% and fix the dimensions K of latent feature vector as 10. For MovieLens 20M dataset, we perform a group of experiments by changing β from 0.1 to 2.9. For Book-Crossing dataset, we perform a group of experiments by changing β from 3 to 6.

The results of changing are shown in Figure 3. MAE decreases along with the value of β increase, MAE is the smallest when β reaches 1.3/4.5, and MAE increases along with the value of β increase when it is beyond 1.3/4.5. It illustrates that only using user-item rating matrix by abandoning tagging information or excessively using tagging information cannot achieve reliable recommendation.
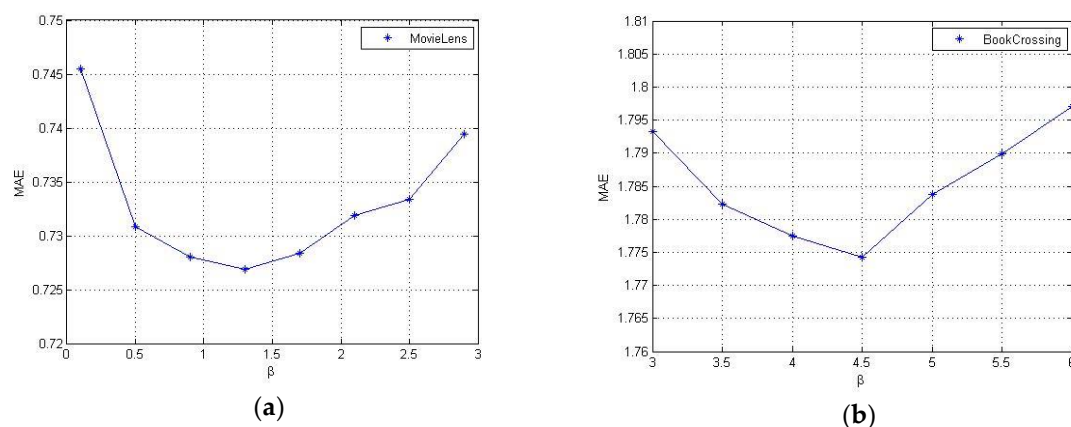
Figure 3. The impact of tagging information: (**a**) the impact of β in MovieLens 20M; and (**b**) the impact of β in Book-Crossing.

3.5. The influence of dimension of latent feature K

The dimensions K of latent feature vectors is another parameter. In this section, based on training set of size 20%, we perform the experiments of TagiCoFi and ETIMF with K from 10 to 50. For MovieLens 20M dataset, the β of TagiCoFi is set to 0.1 while the β of ETIMF is set to 1.3. For Book-Crossing dataset, the β of TagiCoFi is set to 0.1 while the β of ETIMF is set to be 4.5. The results of TagiCoFi and ETIMF with different K are shown in Fig.3. As K increases, the values of MAE decreases. As we known, the larger is the value of K, the more preference can be represented by the latent features. However, Figure 4 show that the improvement gets smaller when the value of K continually increases. It illustrates that the existing latent features can represent useful information when the value of K reaches a certain threshold, and the value of K passing the threshold may introduce noise into the loss function. Form the experimental result, we can observe that ETIMF can gain good performance with K taking a large range of values.
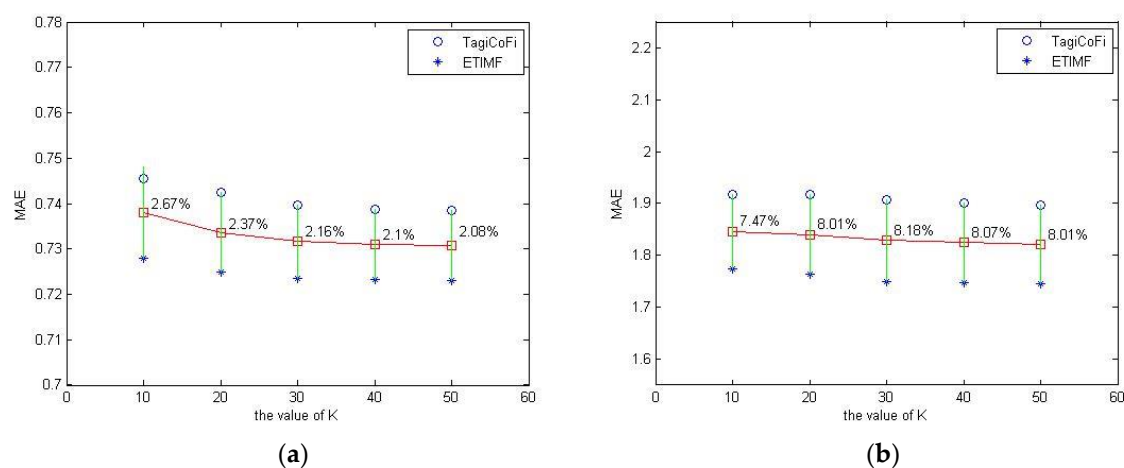

Figure 4. The impact of latent factor number K: (**a**) the impact of K in MovieLens 20M; and (**b**) the impact of K in Book-Crossing.

3.6. Efficiency comparisons

The MAE of ETIMF starts better than that of TMT when the dimensions K of latent feature vectors in ETIMF is equal to 50. Therefore, based on the training set of size 20%, K is set to 50 in MF, TagiCoFi and ETIMF to compare the efficiencies of MF, TagiCoFi,TMT and ETIMF. In the group of

experiments, the values of β in TagiCoFi and ETIMF are set to 0.1 and 1.3, respectively. For MovieLens 20M dataset, the values of β in TagiCoFi and ETIMF are set to 0.9 and 4.5, respectively.

The statistic of running time (the unit is second) of recommendation methods is shown in Table 7. For the training set with the same size, no matter MovieLens 20M dataset or Book-Crossing dataset, the running time of ETIMF is larger than that of MF and that of TagiCoFi, but it is much smaller than that of TMT, which illustrates that the efficiency performance of ETIMF is better than that of TMT. Although the efficiency of ETIMF is worse than that of MF and that of TagiCoFi, the MAE of ETIMF is lowest in all comparison methods.

Table 7. The statistic of running time in two datasets.

Training set	MovieLens 20M				Book-Crossing			
	MF	TagiCoFi	TMT	ETIMF	MF	TagiCoFi	TMT	ETIMF
20%	657	650	805	**648**	199	198	362	**198**
50%	971	969	1244	**978**	634	632	893	**634**
80%	1108	1104	1488	**1127**	1197	1189	1588	**1199**

3.7. Performance of sparse tagging information

Tag sparsity usually has two representations, one is users only annotate a few items with tags, and the other is only a small number of items are annotated with tags. In the case of tag sparsity, the tags annotated to items cannot represent the characteristics of items clearly. Hence, the similarities of items only measuring existing tags of items are inaccurate. In this section, we perform the experiments of TagiCoFi and ETIMF by using a part of tags (10%, 30%, 50% and 80%) randomly selected from existing tags. In this group of experiments, the value of K is set to be 50 uniformly in TagiCoFi and ETIMF. For MoiveLens 20M dataset, the values of β in TagiCoFi and ETIMF are set to 0.1 and 1.3, respectively. For Book-Crossing dataset, the values of β in TagiCoFi and ETIMF are set to 0.9 and 4.5, respectively.

Table 8 and Table 9 show the impact of tagging information size for 20% MovieLens 20M training set and that for 20% Book-Crossing training set respectively. From these results, we can see that the performance of ETIMF is obviously better than that of TagiCoFi and TMT. Furthermore, the value of MAE in ETIMF has no obvious decrease with sparse tags. The experimental results demonstrate that ETIMF performs better in the case of tag sparsity.

Table 8. The Impact of tag information size in MovieLens 20M.

Algorithm	Tag information size			
	80%	50%	30%	10%
TagiCoFi	0.7402	0.7413	0.7429	0.7445
TMT	0.7323	0.7307	0.7318	0.7326
ETIMF	**0.7283**	**0.7286**	**0.7288**	**0.7292**

Table 9. The Impact of tag information size in Book-Crossing.

Algorithm	Tag information size			
	80%	50%	30%	10%
TagiCoFi	1.9592	1.9735	1.9893	2.0117
TMT	1.7962	1.8432	1.8660	1.9060
ETIMF	**1.7885**	**1.8343**	**1.8752**	**1.8973**

3.8. Performance of item cold start problem

Cold start problem is one of the most difficult challenges of recommendation algorithms. Recommender system are required to recommend items to the user who has not rated any items and recommend the item which has not been rated by any user to users. Most collaborative filtering

recommendation algorithms, such as matrix factorization algorithms, have poor performances in the case of cold start problem due to lack of preference information.

In this section, based on the training set of size 20%, we randomly removed the rating records of 50 and 100 items from the training set. Those removed items are regarded as cold start items, namely new items in the recommender system. In this group of experiments, the value of K is set to be 50 uniformly in TagiCoFi and ETIMF. For MoiveLens 20M dataset, the values of β in TagiCoFi and ETIMF are set to 0.1 and 1.3, respectively. For Book-Crossing dataset, the values of β in TagiCoFi and ETIMF are set to 0.9 and 4.5, respectively.

The performances of TagiCoFi, TMT and ETIMF in cold-start setting for different datasets are shown in Table 10 and Table 11. The performance of ETIMF for two datasets is distinctly better than that of TagiCoFi and TMT. It demonstrates ETIMF can integrate extended tags well to recommend new items to users.

Table 10. MAE comparison in MovieLens 20M cold-start setting.

Type	50 cold-start items			100 cold-start items		
	TagiCoFi	TMT	ETIMF	TagiCoFi	TMT	ETIMF
Cold-start items	1.0373	0.8983	0.8439	1.1728	0.9903	0.8851
All items	**0.9678**	**0.8123**	**0.7729**	**1.1335**	**0.8603**	**0.8205**

Table 11. MAE comparison in Book-Crossing cold-start setting.

Type	50 cold-start items			100 cold-start items		
	TagiCoFi	TMT	ETIMF	TagiCoFi	TMT	ETIMF
Cold-start items	2.1477	2.0606	2.0571	2.2643	2.1953	2.1552
All items	**1.9137**	**1.7299**	**1.7023**	**2.1456**	**2.0647**	**2.0223**

4. Discussion and Conclusions

An extended-tag-induced matrix factorization method is proposed for recommender systems in this paper. The proposed method makes a pair of item latent feature vector as similar as possible if they are annotated as similar tags according to extended tags. An item relationship regularization term based on extended tags is added into matrix factorization to constrain the baseline matrix factorization. Experimental results on real datasets demonstrate the proposed algorithm can outperform state-of-the-art collaborative filtering algorithms, including some recommendation methods which use tagging information. Furthermore, it can not only cope with the cold start item problem but also alleviate tag sparsity.

Tagging information not only contains the description of items but also the sentiment of users. For improving the development of recommendation techniques, our future work is to integrate tagging information into recommendation algorithms. One of our future research directions is filling the missing ratings by tagging information before learning latent features. Moreover, we plan to analyze the sentiment of tags to discard noisy tags and classify the remaining tags precisely. More extra information, such as social behaviors of users and social information, will be considered to combine with tag information to provide more precise user profile and item profile to recommender system.

Author Contributions: H.H., M.H., and U.A.B. conceived the algorithm, prepared the datasets, and wrote the manuscript. H..H., and Y.Z. designed, performed, and analyzed the experiments. All authors read and approved the final manuscript.

References

1.   Karabadji N E I, Beldjoudi S, Seridi H. Improving Memory-Based User Collaborative Filtering with Evolutionary Multi-Objective Optimization. *Expert Systems with Applications*, **2018**, *98*, pp. 153-165.

2.   Morozov S 1. An Empirical Study of the Recursive Input Generation Algorithm for Memory-based Collaborative Filtering Recommender Systems. *International Journal of Information & Decision Sciences*. **2017**, *5*, pp. 36-49.

3.   Hu Y, Shi W, Li H. Mitigating Data Sparsity Using Similarity Reinforcement-Enhanced Collaborative Filtering. *Acm Transactions on Internet Technology*, **2017**, *17*, pp. 1-20.

4.   Zhang Z, Kudo Y, Murai T. Neighbor selection for user-based collaborative filtering using covering-based rough sets. *Annals of Operations Research*, **2017**, *256*, pp. 359-374.

5.   Colace F, Santo M D, Greco L, et al. A collaborative user-centered framework for recommending items in Online Social Networks. *Computers in Human Behavior*, **2015**, *51*, pp. 694-704.

6.   Moscato V, Picariello A, Rinaldi A M. Towards a user based recommendation strategy for digital ecosystems. *Knowledge-Based Systems*, **2013**, *37*, pp. 165-175.

7.   Guo G, Zhang J, Zhu F, et al. Factored similarity models with social trust for top-N item recommendation. *Knowledge-Based Systems*, **2017**, *122*, pp. 17-25.

8.   He J, Liu H, Xiong H. SocoTraveler : Travel-package recommendations leveraging social influence of different relationship types. *Information & Management*, **2016**, *53*, pp. 934-950.

9.   Albanese M, Chianese A, D'Acierno A, et al. A multimedia recommender integrating object features and user behavior. *Multimedia Tools & Applications*, **2010**, *50*, pp. 563-585.

10. Yin H, Cui B, Zhou X, et al. Joint Modeling of User Check-in Behaviors for Real-time Point-of-Interest Recommendation. A*cm Transactions on Information Systems*, **2016**, *35*, pp. 11.

11. Manotumruksa J, Craig Macdonald, Ounis I. Regularising Factorised Models for Venue Recommendation using Friends and their Comments. International Conference on Information and Knowledge Management. Indianapolis, IN, USA, 24-28 October, 2016; pp. 1981-1984.

12. Kharrat F B, Elkhleifi A, Faiz R. Recommendation system based contextual analysis of Facebook comment. International Conference of Computer Systems and Applications. Agadir, Morocco, 29 November - 2 December 2017; pp. 1-6.

13. Wang C, Wang C, Wang H, et al. Attributes coupling based matrix factorization for item recommendation. *Applied Intelligence*, **2017**, *46*, pp. 1-13.

14. Gedikli F. Improving recommendation accuracy based on item-specific tag preferences. *ACM Transactions on Intelligent Systems and Technology (TIST)*, **2013**, *4*, pp. 1-19.

15. Chatti M A, Dakova S, Thus H, et al. Tag-Based Collaborative Filtering Recommendation in Personal Learning Environments. *IEEE Transactions on Learning Technologies*, **2013**, *6*, pp. 337-349.

16. Peng J, Zeng D. Topic-based web page recommendation using tags. In Proceedings of the IEEE International Conference on Intelligence and Security Informatics. Dallas, Texas, USA, 8-11 June 2009; pp. 269-271.

17. Blaze J, Asok A, Roth T L. Content-based tag propagation and tensor factorization for personalized item recommendation based on social tagging. *Acm Transactions on Interactive Intelligent Systems*, **2014**, *3*, pp.26.

18. Tso-Sutter K H L, Marinho L B, Schmidt-Thieme L. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In Proceedings of the ACM Symposium on Applied Computing, Fortaleza, Ceara, Brazil, 16-20 March 2008; pp. 1995-1999.

19. Zhen Y, Li W J, Yeung D Y. TagiCoFi: tag informed collaborative filtering. In Proceedings of the ACM Conference on Recommender Systems, New York, USA, 22-25 October 2009; pp. 69-76.

20. Huang C L, Lin C W. Collaborative and content-based recommender system for social bookmarking website. *World Academy of Science Engineering & Technology*. **2010**, *68*, pp. 748.

21. Kim H N, Ji A T, Ha I, et al. Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation. *Electronic Commerce Research & Applications*. **2010**, *9*, pp.73-83.

22. Nguyen J, Zhu M. Content-boosted matrix factorization techniques for recommender systems. *Statistical Analysis & Data Mining the Asa Data Science Journal*. **2013**, *6*, pp. 286-301.

23. Huang C L, Yeh P H, Lin C W, et al. Utilizing user tag-based interests in recommender systems for social resource sharing websites. *Knowledge-Based Systems*. **2014**, *56*, pp. 86-96, 2014.

24. Rawashdeh M, Kim H N, Alja'Am J M, et al. Folksonomy link prediction based on a tripartite graph for tag recommendation. J*ournal of Intelligent Information Systems*, **2013**, *40*, 307-325.

25. Kim H N, Rawashdeh M, Saddik A E. "Tailoring recommendations to groups of users", In Proceedings of the International Conference on Intelligent User Interfaces, Santa Monica, CA, USA, 19-22 March 2013; pp. 15-24.

26. Ji K, Shen H. Addressing cold-start: Scalable recommendation with tags and keywords. *Knowledge-Based Systems*. **2015**, *83*, pp. 42-50.

27. Fang Z, Gao S, Li B, et al. Cross-Domain Recommendation via Tag Matrix Transfer. In Proceedings of the IEEE International Conference on Data Mining Workshop, Atlantic City, New Jersey, USA, 15-17 November 2015; pp. 1235-1240.

28. Paterek A. Improving regularized singular value decomposition for collaborative filtering. In the Proceedings of Kdd Cup & Workshop, San Jose, California, USA, 12 Aug 2007.

29. Luo X, Xia Y, Zhu Q. Incremental Collaborative Filtering recommender based on Regularized Matrix Factorization. *Knowledge-Based Systems*. **2012**, *27*, pp. 271-280.

30. Dong X, Thanou D, Frossard P, et al. Learning Laplacian Matrix in Smooth Graph Signal Representations, *IEEE Transactions on Signal Processing*. **2014**, *64*, pp. 6160-6173.

31. Nemirovski A, Juditsky A, Lan G, et al. Robust Stochastic Approximation Approach to Stochastic Programming. *Siam Journal on Optimization*. **2014**, *19*, pp. 1574-1609.

32. http://grouplens.org/datasets/movielens/20m/

33. Ziegler C N, Mcnee S M, Konstan J A, et al. Improving recommendation lists through topic diversification. In Proceedings of the International Conference on World Wide Web, Chiba, Japan, 10-14 May 2005;pp. 22-32.

34. Koren Y, Bell R, Volinsky C. Matrix Factorization Techniques for Recommender Systems. *Computer*. **2009**, *42*, pp. 30-37.