Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2022) 30: 109 – 126

doi:10.3906/elk-2008-26

Improving collaborative recommendation based on item weight link prediction

Sahraoui KHARROUBI[1,*], **Youcef DAHMANI**[1], **Omar NOUALI**[2]
[1]department of Computer Science, Faculty of Mathematics and Computer Science ,
Ibn Khaldoun University, Tiaret, ALGERIA
[2]Basic Software Laboratory, C.E.R.I.S.T., Ben Aknoun, Algiers, AlGERIA,

Received: 06.08.2020 • **Accepted/Published Online:** 10.08.2021 • **Final Version:** 19.01.2022

Abstract: There is a continuous information overload on the Web. The problem treated is how to have relevant items (documents, products, services, etc.) at time and without difficulty. Filtering system also called recommender systems are widely used to recommend items to users by similarity process such as Amazon, MovieLens, Cdnow, etc. In the literature, to predict a link in a bipartite network, most methods are based either on a binary history (like, dislike) or on the common neighbourhood of the active user. In this paper, we modelled the recommender system by a weighted bipartite network. The bipartite topology offers a bidirectional reasoning item side and user side, which preserve the information shared between the nodes. To make such a prediction, we seek to determine the shares of items shared between users. In the first step, we accumulate the shares of the users towards the items and in the second step, the shares of the items towards the users. The idea is to exploit the item-user connectivity to predict nonexistent links based on existing links. Therefore, the information is propagated linearly and without loss between nodes in the network. We focused on the item through a dual projection that combines the sharing of quotas between users and items. Empirical tests on real data sets (Amazon-Books, MovieLens 1M, Yelp2018, Yahoo! Songs) showed satisfactory results.

Key words: Information propagation, recommender systems, ranking, item-user, weighted network

1. Introduction

In a few years, 95% of purchases will be made online. Access to relevant items (documents, movies, hotels, etc.) becomes more difficult because, despite their availability, they are lost in the mass [1]. Recommendation systems have been widely used by companies to recommend relevant items to users by employing a similarity process. Large companies and websites such as Netflix, Amazon, Facebook, YouTube, Twitter, etc. integrate the techniques of recommendation in their servers. The methods of recommendation systems can be mainly classified into three approaches [2]: the content-based approach which compares the content of the user profile with the content of the item; the collaborative approach which depends on the feedback of neighbour's ratings; and the hybrid approach which aims at objectively combining the two methods. The collaborative approach is the most widely used and effective in many areas [3, 4]. However, finding the nearest neighbours is the critical phase of the collaborative approach. In addition, providing high-quality recommendations to users with a minimum of common feedback is a major challenge for recommendation systems [5, 6]. Currently, several in-depth studies are focused on modelling using bipartite networks [7, 8]. The structure of bipartite networks is perfectly adapted as a theoretical and practical model for several systems in the real world. The recommen-

*Correspondence: sahraoui.kharroubi@univ-tiaret.dz

dation process in e-commerce or e-learning is modelled by a bipartite network that links two distinct sets of users and items. The recommendation task in the recommendation system is equivalent to a link prediction in a bipartite network. What is the probability of having a link between two nodes (user, item) at time t + 1 knowing existing links at time t. Link prediction is a subject of intensive research in recommendation systems and social networks to determine new users [9]. This paper, which draws on relevant works [10, 11], presents a recommendation method that is the basis of link prediction in a weighted bipartite network. We propose a Item Share propagation algorithm for Link Ranking in recommender systems *(ISpLR)*. The method aims to determine the weights of a link as a ranking value predicted by the user towards an item. The idea is to accumulate the acquired information by linking the items to the users through forward-backward projection. The process can be summarized as follows:

(a) The proposal of a method called Item Share propagation for Link Ranking *(ISpLR)*, which focuses on the item to calculate the nonexistent link weight in a weighted bipartite network.

(b) The weight of a nonexistent link is calculated from an accumulation of the user's shares and the item's shares with dual projection in the network. However, the recommendation decision is based on the weight of this link.

(c) Implementation of the method *(ISpLR)* without adjustment parameters and with different dense datasets.

This paper is organized as follows: the background is given in Section 2, a detailed description of the method is presented in Section 3, a description of the experimental phase and discussion of the results is given in Sections 4 and 5, respectively. In the conclusion section, we make some suggestions for future works.

2. Background

Collaborative filtering (CF) is the most widely used and effective approach to create personalized recommendations on the Web [12]. The crucial task of CF is to form communities (neighbourhoods) of users or items through similar feedback. The community is used to predict items deemed relevant to users who have not yet judged these items. Over the past two decades, several algorithms [13–15], have been proposed and these can be classified into three main levels.

Traditional approaches

They are based on the rating history of users «rating matrix»and apply statistical methods to make predictions. The assumption is that users with similar ratings would have similar tastes in the future. These approaches are divided into two categories: memory-collaborative filtering (memory CF) and model-collaborative filtering (model CF) [16]. Memory CF uses the set of rating matrix scores to make predictions about the active user. The algorithm starts by calculating the similarity between users or items by a cosine or Pearson correlation measure or by any other distance measure. Then it selects the active user's neighbourhood (top similar neighbours) to calculate the prediction value of an item. Finally, the algorithm compares the prediction value to a given threshold for the recommendation decision [17]. In order to reduce the computational complexity, model CF uses a part of the «rating matrix»to estimate or learn a model that generates the predictions. Among them, the latent factor model (LFM) is very competitive and widely adopted for the implementation of recommendation systems [18].

Context-aware and semantic approaches

Context is additional information categorized in dimensions to better describe the user or item. These dimensions include personal information (e.g., gender, occupation, age), professional and social information (activities, interests, interactions with other users, preferences, etc.), physical conditions (weather, noise, etc.), location (geographic coordinates), and time. The objective is to value this information and gather similar contexts to predict an item. Adomavicius and Tuzhilin (CARS context-aware recommender systems) [19], using the mobile phone and the Global Positioning System (GPS). CARS can recommend a restaurant or hotel to a tourist based on previous choices of its similar neighbours by context. In addition, it is possible to use the context [20] in collaborative filtering in the absence of the ratings and overcome the problem of the sparse matrix [16]. Thus, several authors [21] use the ontologies and semantic web layers [22] to improve the performance and accuracy of the recommendation. Ontologies make inferences via a reasoning mechanism and extracting unambiguous information.

Graph-based approaches

Each node is linked to another node forming a network structure with a particular topology. The unipartite topology represents a set of nodes and a set of edges as in the case of social networks such as Facebook, Twitter, YouTube, etc. PageRank [11] and HITS [23] are the most important algorithms for calculating the importance score of a node by a random walk or an iterative process in a network of web pages. The EdgeRank algorithm fills [24] a user's Facebook profile through the parameters of affinity, type, and freshness. The goal is to exploit the network structure to collect possible links in the future. Several studies [10] are based on a bipartite structure to improve recommendation algorithms. Nesreen K. [25] implemented the *SimAdapt* algorithm to measure the similarity between two nodes according to the number of common neighbours in a bipartite graph. The research presented by Jung-Hun Kim et al. [26] exploits a bipartite network between textual titles and keywords in a research article in the field of physics. Other works [27, 28] exploit bipartite network topologies to calculate the degree of similarity between users according to two approaches: local and global. The first is the *Node Neighbourhoods* topological approach where the link prediction between two users is based on the number of common neighbours. The measures used in this approach include common neighbours, Adamic/Adar index, preferential attachment and Jaccard coefficient. The second one which is the path topological approach or the prediction of a link between two users is based on the length of the path between them. The measures used include *Katz*, *Hitting Time*, and *Rooted PageRank*. This work focuses only on binary bipartite graphs (like, dislike), whereas admiration can be expressed on a discrete scale where link weighting is involved.

This study applies filtering techniques to generate new links between the two sets as new research hypotheses. A major portion of this research addresses the problem of link prediction in unweighted bipartite networks. The paper presents a method for recommending items in a weighted bipartite network. Based on the item-rating history, the method is used to predict new links and recommend relevant items to users. For this, a cumulative share is counted on the sides of both items and users.

3. Approach

3.1. Problem and motivation

Many complex systems are based on a network model that consists of two components: nodes and links. These systems focus mainly on networks with single-party topology (social networks, links between web pages, atoms and molecules etc.). A single-party network consists of a single set of nodes that are connected by links. Network model algorithms are easy to interpret, implement, and operate. A recommendation system is a typical case of a particular network called the bipartite network. So, how can the components of a recommendation system be modelled according to a bipartite network, and what is the solution to improve the recommendation quality? These questions lead us to the link prediction problem, which is used to find the probability of a link at time t+1 based on the links existing at time t. Our objective is to go further to find the weight of nonexistent links according to the weights of existing links in a bipartite network. The recommendation system is based on this weight to recommend or reject an item to a user.

3.2. Modelling with a weighted bipartite network

A network is bipartite if the set of nodes is divided into two distinct sets I and U. Each link has one end in the first set and the other end in the second set (no links in the same set), and $R(i,j)$ determines the weight of the link. Formally, let $\mathfrak{B}(N,L)$ be a bipartite network where N is the set of nodes and L is the set of links. $R(i,j)$ represents the weight of the link (i,j). $\mathfrak{B}$ has the following properties:

$$N = I \cup U$$
$$L = \{(i,u,R(i,j)), i \in I, u \in U, R(i,j) \in \mathbb{R}^+\}$$
$$I \cap U = \varnothing.$$

A weighted network refers to a network whose links are quantified. Each link has a value that represents the degree of the relationship between the two nodes. The topology of a weighted bipartite network accurately reflects the architecture of a recommender system. In particular, the collaborative filtering approach that is widely adopted in recommender systems is perfectly modelled by this type of network. The items (movies, books, web pages, etc.) represent the nodes of the first set, the users represent the nodes of the second set, and the link value represents a rating value.

3.3. Item link prediction

Several researches [11, 23, 28] address the problem of link prediction around the user node according to the unweighted links of its neighbours. However, the link expresses a binary action (purchase, likes, positive sign), the absence of a link expresses the negation (no-purchase, dislike, negative sign). The prediction of a link to the active user depends directly on the links of its neighbours. *SibRank* [29] based on two types of links: positive if there is an agreement between the user and the preference or negative if not. To infer the similarity between users, the method explores the multiplicative propagation of trust/distrust signs according to the principle «the enemies of my enemy and the friends of my friend are friends, while the enemies of my friend and the friends of my enemy are enemies ». The *Srank* function makes it possible to estimate the agreement or disagreement between the active user and the others through the information deduced from the structure of the graph. In this work, we present a new link prediction method based on the item approach in a weighted bipartite network. The weight of a link is expressed on a numerical rating value more or less precise than a binary value. The method is based on the preservation of hidden information (interestingness) through connectivity weighted in dual projection items/users.

Consider a recommender system which is composed of a set of items $\{I_1, I_2, I_N\}$ and a set of users $\{U_1, U_2, U_M\}$ linked by weighted links $\{(i,u,R(i,j)), i \in I, u \in U, R(i,j) \in \mathbb{R}^+\}$. From the previous interactions of users with the items (the rating history), let us try to deduce a nonexistent link weight. In another way, can we recommend a relevant planned item to a user? This decision is equivalent to valuing the link weight between the item and the user, like (I_3, U_1) (Figure 1).

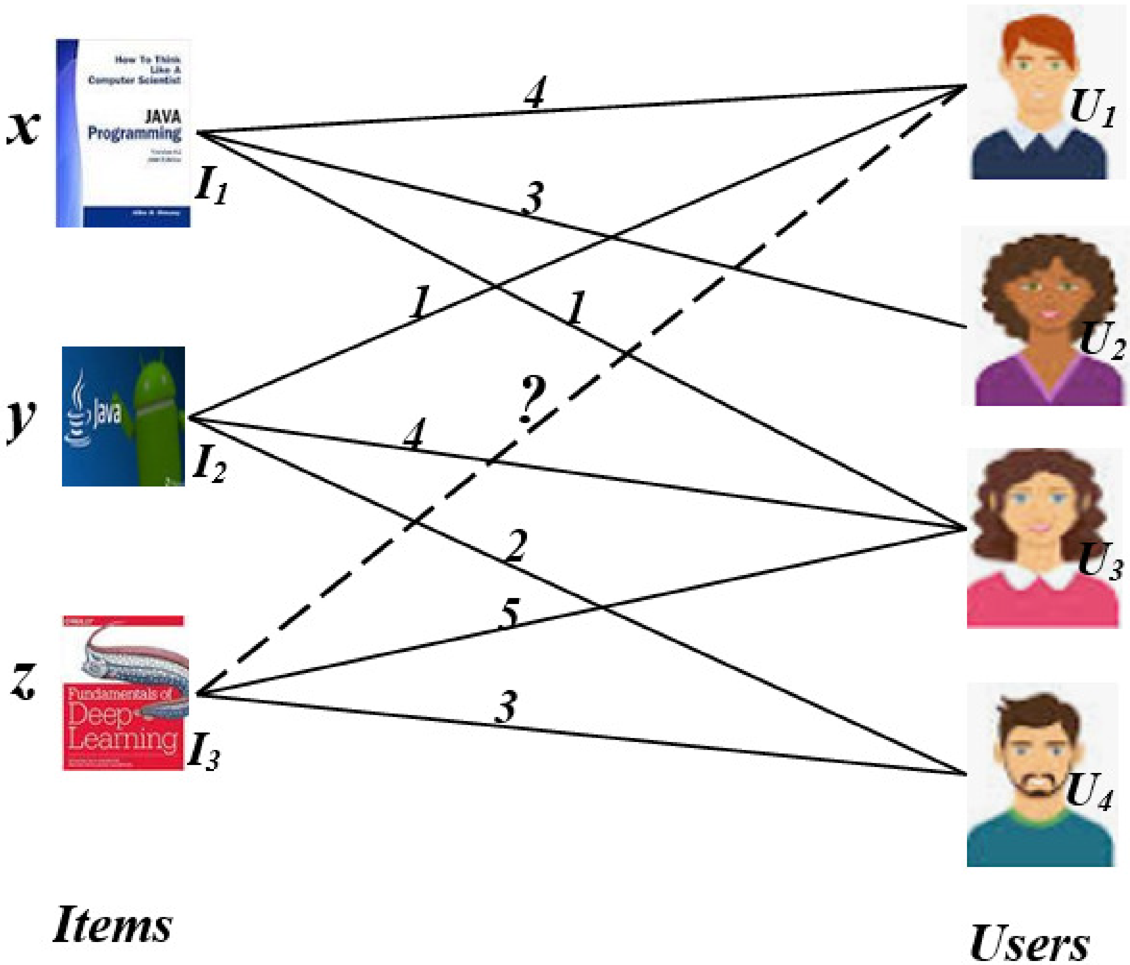

Figure 1. Weighted link prediction.

To make such a prediction we seek to determine the shares of items shared between users. In the first step (forward step), we accumulate the quotas of the users towards the items and in the second step (backward step) the shares of the items towards the users, knowing that N is the number of items, M is the number of users, and $R(i,j)$ is the weight of the link (i,j) (rating value).

In the first step, each item has a cumulative of ratings,

$$IC_i = \sum_{j=1}^{M} R(i,j). \tag{1}$$

The share of user j,

$$q(j) = \sum_{i=1}^{M} \frac{R(i,j)}{IC_i}. \tag{2}$$

In the second step, each user has a cumulative of ratings,

$$UC_j = \sum_{i=1}^{N} R(i,j). \tag{3}$$

The share of item i,

$$q(i) = \sum_{j=1}^{M} \frac{R(i,j)}{UC_j}. \tag{4}$$

Then, the value of link prediction of the item t to the user s is given by the following formula:

$$P(t,s) = \sum_{j=1}^{M} \frac{R(t,j)}{\sum_{i=1}^{N} R(i,j)} * (\sum_{i=1}^{N} R(i,j)) * \frac{R(i,s)}{\sum_{j=1}^{M} R(i,j)}. \tag{5}$$

By replacing item cumulate (1) and user cumulate (3), we get

$$P(t,s) = \sum_{j=1}^{M} \frac{R(t,j)}{UC_j} * (\sum_{i=1}^{N} R(i,j)) * \frac{R(i,s)}{IC_i}. \tag{6}$$

The bipartite topology offers a bidirectional reasoning item side and user side, which preserve the information shared between the nodes. The consideration of the weights of the links specifies the degree of satisfaction of the users. This method presents an item-based prediction, but the same user-based reasoning can also be applied (Figure 2).

The example below details the method proposed via the two variants namely item link prediction and user link prediction, specifying the prediction performance between them. Based on the link values presented in Figure 1:

a) Item link prediction

Either three items I_1, I_2, I_3 with initial inputs x, y, z, respectively.

First step (forward): Determining user's share,

$$\begin{cases} U_1 : \frac{4}{8}x + \frac{1}{7}y \\ U_2 : \frac{3}{8}x \\ U_3 : \frac{1}{8}x + \frac{4}{7}y + \frac{5}{8}z \\ U_4 : \frac{2}{7}y + \frac{3}{8}z \end{cases}$$

Second step (backward): Determining item's share,

$$\begin{cases} I_1 : \frac{4}{5}U_1 + \frac{3}{3}U_2 + \frac{1}{10}U_3 + \frac{0}{5}U_4 \\ I_2 : \frac{1}{5}U_1 + \frac{0}{3}U_2 + \frac{4}{10}U_3 + \frac{2}{5}U_4 \\ I_3 : \frac{0}{5}U_1 + \frac{0}{3}U_2 + \frac{5}{10}U_3 + \frac{3}{5}U_4 \end{cases}$$

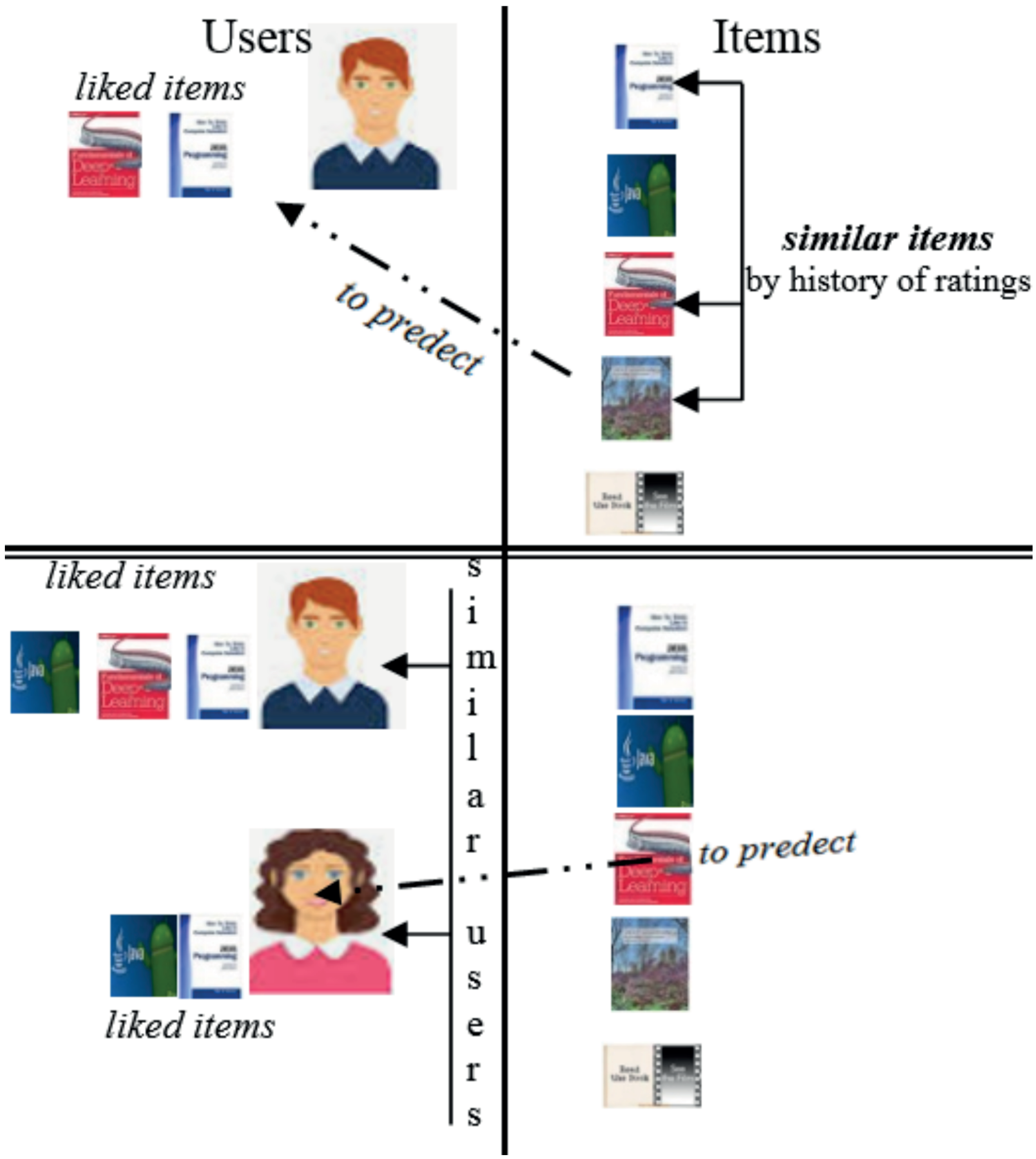

Figure 2. Item link prediction versus user link prediction.

Which gives us:

$$\begin{cases} I_1 : \dfrac{63}{80}x + \dfrac{6}{35}y + \dfrac{1}{16}z \\ I_1 : \dfrac{3}{20}x + \dfrac{13}{35}y + \dfrac{2}{5}z \\ I_1 : \dfrac{1}{16}x + \dfrac{16}{35}y + \dfrac{43}{80}z \end{cases}$$

Table 1 shows the item link predictions.

So, the prediction value of the $link(I_3, U_1) = 1/16\ (4) + 16/35\ (1) + 43/80\ (0) = 0.7071$

Further, the prediction value of the $link(I_1, U_4) = 63/80\ (0) + 6/35\ (2) + 1/16\ (3) = 0.5304$

Table 1. Item link prediction.

	U_1	U_2	U_3	U_4
I_1	4	3	1	**0.5304**
I_2	1	**0.45**	4	2
I_3	**0.7071**	**0.1875**	5	3

To know the error of the prediction, we generalize the calculation for existing vote values (Table 2) and apply the mean absolute error measure MAE (see subsection 4.4).

Table 2. MAE of item-based predictions.

	U_1	U_2	U_3	U_4	$MAE\ I_i$
I_1	**3.3214**	**2.3625**	**1.7857**	**0.5304**	***0.7006***
I_2	**0.9714**	**0.45**	**3.6357**	**1.9429**	***0.1500***
I_3	**0.7071**	**0.1875**	**4.5786**	**2.5268**	***0.4473***
	MAE (mean)				**0.4326**

The MAE measure makes it possible to differentiate between the rating value given by the users and the prediction value generated by the formula (6). For item I_2 and I_3:

$MAE(I_2) = 1/3\ (|1 - 0.9714| + |4 - 3.6357| + |2 - 1.9429|) = 0.1500$

$MAE(I_3) = 1/2\ (|5 - 4.5786| + |3 - 2.5268|) = 0.4473$

b) User link prediction

First step (forward): Determining item's share,

$$\begin{cases} I_1 : \frac{4}{5}x + \frac{3}{3}y + \frac{1}{10}z \\ I_2 : \frac{1}{5}x + \frac{4}{10}z + \frac{2}{5}f \\ I_3 : \frac{5}{10}z + \frac{3}{5}f \end{cases}$$

Second step (backward): Determining user's share,

$$\begin{cases} U_1 : \frac{3}{7}x + \frac{1}{2}y + \frac{3}{28}z + \frac{2}{35}f \\ U_2 : \frac{3}{10}x + \frac{3}{8}y + \frac{3}{80}z \\ U_3 : \frac{3}{14}x + \frac{1}{8}y + \frac{31}{56}z + \frac{169}{280}f \\ U_4 : \frac{2}{35}x + \frac{169}{560}z + \frac{19}{56}f \end{cases}$$

Similarly, Table 3 shows the MAE of user-based predictions.

We can clearly see that the item-based predictions (MAE = 0.4326) are more precise than the user-based predictions (MAE = 0.4450).

Table 3. MAE of user-based predictions.

	I_1	I_2	I_3	$MAE\ U_i$
U_1	**3.3214**	**0.9714**	**0.7071**	***0.3536***
U_2	**2.3625**	**0.4500**	**0.1875**	***0.6375***
U_3	**1.7857**	**3.6357**	**4.5786**	***0.5238***
U_4	**0.5304**	**1.9429**	**2.5268**	***0.2652***
	MAE (mean)			**0.4450**

3.4. Recommendation task

A recommender system processes a large number of items and selects the most relevant to recommend them to users. The decision to recommend a particular item depends on the predicted value deduced by the system. This value is compared to a δ threshold as an input parameter of the algorithm. The recommendation decision of an item t to a user s $R(t,s)$ is given by the formula below:

$$R(t,s) = \begin{cases} t \quad \textit{recommended to s} & \text{if } p(t,s) >= \delta \\ t \quad \textit{not recommended to s} & \text{otherwise} \end{cases} \tag{7}$$

$p(t,s)$: prediction value of item t to user s.

The threshold δ is bound to the rating scale as a degree of user satisfaction, for example for MovieLens and Amazon, the rating scale is limited to 5. Algorithm 1 summarizes the main sequences of a recommendation task by ISpLR.

Algorithm 1 *Item link recommendation*

input *Bipartite Network* $\mathcal{B}(\mathcal{N}, \mathcal{L})$
    $\mathcal{N} = I \cup U$ *set of nodes*
    $\mathcal{L} = \{(t,s), t \in I, s \in U\}$ *set of links*
    $\mathcal{W} : \mathcal{L} :\rightarrow \mathbb{R}^+$ *weighted function*
    $(t,s) \rightarrow R(i,j)$
output *Weight link*
  $w(t,s) = 0$;
  For j=1 to M
   $UC_s = 0$;
   For i=1 to N
    $UC_s = UC_s + R(i,j)$;   % s^{th} *user cumulate*
   $q_s = 0$;
   For i=1 to N
    $IC_t = 0$;
     For k=1 to M
      $IC_t = IC_t + R(i,k)$;

$$q_s = q_s + \frac{R(i,j) * R(i,s)}{IC_t}; \qquad \%\ t^{th}\ \ item\ cumulate$$

$$w(t,s) = w(t,s) + \frac{R(t,j) * q_s}{UC_s};$$

$Return\ \ w(t,s)$;

if $w(t,s) >= \delta$ $(recommend\ t\ to\ s)$;

This item-based method preserves a propagation of shared information between the two types of nodes (items and users). The recommendation task is modelled according to weighted bipartite network topology.

Often, the user space is larger than the item space, which favours the item-based method. Therefore, calculating similarities between users is more expensive than calculating similarities between items. In addition, the latter can often be done offline and in a timely manner, as is the case for e-commerce sites.

4. Experimentation

4.1. Datasets

To evaluate the effectiveness of the proposed approach, tests were performed on four real-world datasets:

- **Amazon-Books:** a free download dataset http://jmcauley.ucsd.edu/data/amazon/for research purposes. Amazon is an e-commerce recommendation system for a wide range of products. Amazon datasets are widely used for empirical testing, particularly in the recommendation domain. We have selected the dataset from the "Books" category where the entries are in the form of triplets ‹*ID user, ID item, review*›.

- **MovieLens:** a movie evaluation dataset https://grouplens.org/datasets/movielens/ provided by the GroupLens research group. The datasets are used for research purposes, in particular, to test the performance of collaborative filtering algorithms. There are several versions of the datasets ranging from 100K to 20M evaluations according to defined evaluation periods.

- **Yelp2018 dataset:** a set of data provided by the Yelp recommendation platform https://www.yelp.com/dataset. It includes statistics and user ratings for different items such as restaurants, hotels, shopping centers, etc. The latest version contains more than 8 million ratings rated by more than 1 million users.

- **Yahoo! Song:** a dataset for user ratings of music tracks and albums. This dataset is provided by the Yahoo Research Alliance's Webscope program http://webscope.sandbox.yahoo.com/. It includes the files trainldx, trackData, albumDatat, artistData where we focused on idItem, Iduser, and the rating score.

4.2. Experimental process

The first step was to extract necessary entries from downloaded files and datasets. For Amazon Book 5-core, the dataset contains 22,507,155 reviews, 8,026,324 users and 2,370,585 books where each user or item has at least 5 interactions. The attributes *reviewer ID* (user identifier), *ASIN* (book identifier), and *overall* (book rating) are selected from user review data file and product review data file. For MovieLens dataset, we selected the 1M version containing 6,040 users, 3,706 movies and 1,000,209 reviews where each user rated at least 10 movies. The triplets $\langle userID, movieID, rating\ \rangle$are retained from the *ratings.csv* file. Yelp dataset includes

409,117 users, 85,539 items and 2,685,066 ratings. The attributes *business id*, *user id*, and *stars* are extracted from *review.csv* file with 10-core setting for better data quality. Similarly, track ratings are taken from *Yahoo! Music User Ratings* of Musical Artists, version 1.0. Each set of data is divided into two parts: learning set, and testing set. The file formats (*csv*, *json*) are converted to MATLAB code in matrix form. Items corresponding to the first set of nodes and users correspond to the second set of nodes of the two-party network. The rating scores represent the weights of the links. Based on the links in the learning set, the algorithm tries to predict the weight of the new links in the testing set.

4.3. Comparison methods

Many studies focus on the evolution of the network and analyze the interactions between its components. Exploiting the links helps to find new friends or new items, which may be important. Traditional collaborative systems [1, 32] are based on correlations between users or items to estimate prediction values. For timely prediction, such as online shopping or instant recommendations, the use of dimension reduction or clustering methods [33, 34] is necessary. To see the effectiveness of the ISpLR proposal, we have implemented three methods related to this work. The experiments are tested on the data sets listed above, which are available for research purposes. After downloading, each dataset was split into two parts one for training (85%) and the other for testing (15%).

Signed bipartite network analysis for neighbor-based collaborative ranking (SibRank) [29]: The authors seek to improve the neighbour-based collaborative ranking approach of users through a signed bipartite network. Between the user's set and preference's set, there are two types of links: positive if there is an agreement between the user and the preference or negative if not. First, the algorithm builds a signed network (*SiBreNet*) made up of all users, all items and the pairs of item preferences. Next, an *SRank* function is designed to aggregate the propagated signs of the target user and his neighbours in order to estimate a preference matrix. As a result, keep only k most similar neighbours to deduce the ranking of the top *N-items* to be recommended. Experimental tests of the algorithm were applied on MovieLens-100K dataset. On our side, we used the *u.DATA* table from the same data set (version 1M) to build a signed preference network. A threshold is applied to estimate a positive sign if the rating is greater than or equal to 3 on a scale of 1 to 5, a negative sign otherwise. From this signed preferences network, we deduce a matrix of similarity between users. The top *N-items* preferences of the most similar users are recommended to the active user.

Neural Graph Collaborative Filtering (NGCF) [30]: In this work, the authors focus on high-order connectivity of user-item interactions through a neural graphic model. The process runs through a hierarchy of layers; in the first iteration, it captures the items linked directly to the active user (layer one), in the second iteration the process captures the users linked to these items (layer two), and so on. At each iteration, there is an aggregation of connectivity between users and items. The sum of the aggregation makes it possible to deduce the degree of similarity between users or between items. The authors develop two functionalities: message construction which defines the information propagated from item i to the user u and message aggregation which defines the accumulation of information propagated from the neighbourhood of item i in each layer. The experiments are conducted on three benchmark datasets: Amazon-book, Yelp2018 and Gowalla. In our study, we applied the information shared linearly between two layers with the integration of link weights and not only the connectivity between the user and the item. Moreover, this approach suffers from the problem of scalability if the number of layers is high enough.

Spectral Collaborative Filtering (SpectralCF) [31]: builds a *SpectralCF* recommendation model to deal with the cold start problem. This problem occurs if the user is not very active (low ratings) or if it is a new user. Then the recommendation system does not have enough information to predict good recommendations. On the one hand, *SpectralCF* exploits the proximities of the graph and extracts hidden connectivity information. On the other hand, this study proposes a spectral convolution filter on the vertices (users and items) of the bipartite graph to dynamically filter the contribution of each frequency component in the spectral domain. To evaluate the performance of the ranking, the experiments used two metrics *Recall@M* and *MAP@M* based on MovieLens-1M and *Amazon Instant Video* datasets.

In addition, we have varied our method based on users (USpLR) and seen the results obtained. All the methods are implemented in MATLAB interpreted language according to the datasets mentioned in the previous section.

4.4. Metrics

In the literature on recommendation, several metrics are used such as *MAE* (mean absolute error), *NMAE* (normalized MAE), *MSE* (mean square error), *RMSE* (root MSE) as well as other classification measures such as *recall*, *precision*, *F-measure*, *NDCG* (normalised discounted cumulative gain). The most common being the *MAE* and the *recall*.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|r_{ij} - p_{ij}| \tag{7}$$

r_{ij}: rating of item i by user j;
p_{ij}: prediction of item i by user j;
n: rating number.
recall : the fraction of pertinent returned items by all pertinent items.

$$recall = \frac{N_{pr}}{N_p} \tag{8}$$

User satisfaction is strongly linked to the quality of prediction that is measured by the *MAE* and *recall* metrics.

5. Results and discussion

We divided the datasets into two parts, 85% as a training set and 15% as a test set. The algorithm is based on the training set and generates the predictions of the test set. After a learning phase (on training set) and for each value of the neutralized link weight of the test set, the algorithm is launched to find this missing value. The found value represents a prediction value of the neutralized link weight. The difference between the generated predictions and the real votes of the learning set is measured by MAE according to the number of recommendations. The recommendation parameter (δ) is set to 3 on a scale [1–5], i.e. if the prediction value of the item is greater than 3, then the item is recommended to the user. Figure 3 shows the error rate according to the number of recommendations.

The result obtained by MAE (Figure 3) shows a better performance in favour of the proposed method (ISpLR) compared to SpectralCF, SibRank, and NGCF respectively. We find that the MAE also varies depending on the datasets, which are better in MovieLens and Yahoo! Song and worse in Yelp2018 and Amazon-Book. The reasons for these results are as follows:

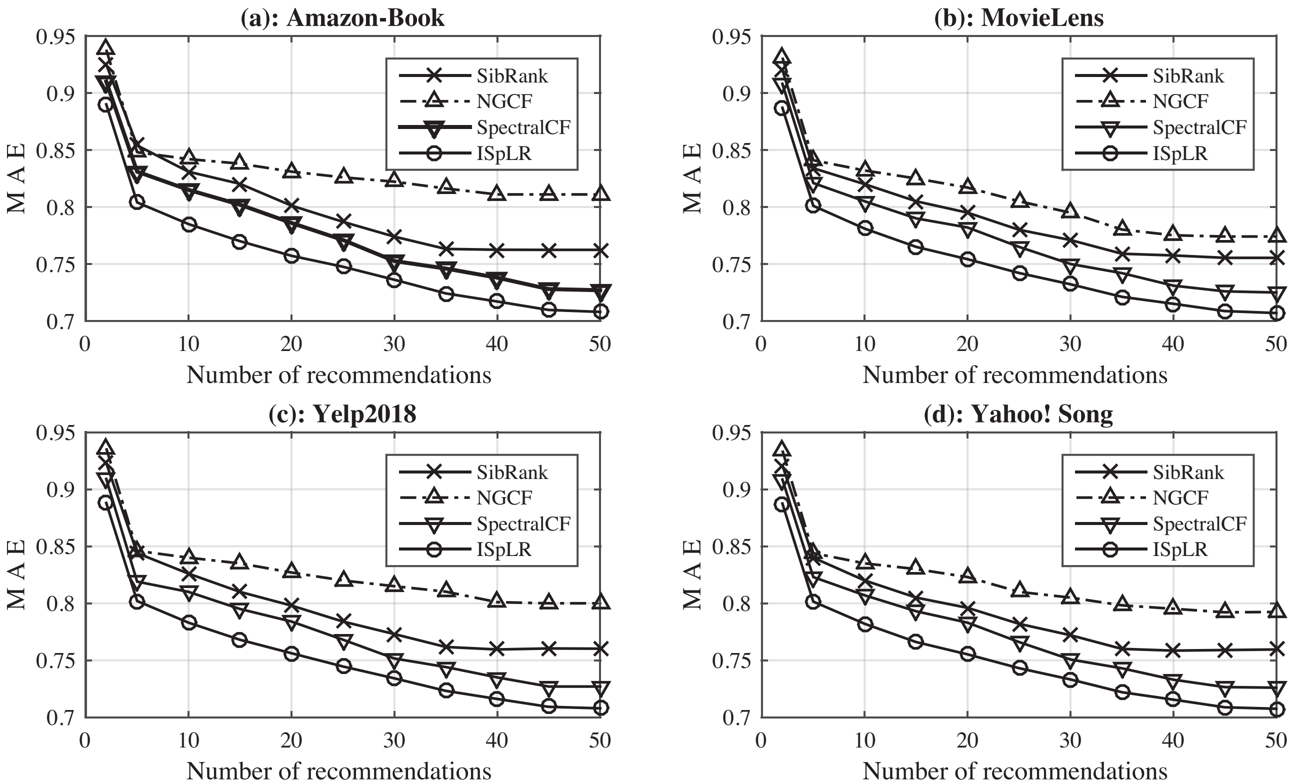

Figure 3. Error rate depending on the length of the recommendation list.

(1) The advantage of ISpLR lies in the preservation of information shared between items and users. Therefore, the information is propagated linearly without loss between each node in the network, which considerably minimizes the error rate. The *SpectralCF* method cumulates the hidden connectivity of the proximity nodes which slightly reduces the error;

(2) The different datasets used have an impact on the MAE according to the sparsity rate (formula 9) of each dataset as shown in Table 4.

$$sparsity\ rate = 1 - \frac{|E|}{|I| * |U|}, \tag{9}$$

where $|E|,|I|,|U|$ is the number of edges, items and users respectively.

Table 4. Sparsity rate.

Dataset	MovieLens 1M	Yahoo!Song	Yelp2018	Amazon Books
Sparsity rate	95.754%	97.976%	99.957%	99.988%

It is clear that the sparsity rate (no rating) is proportional to the error where the quality of the recommendation system is strongly related to the number of judgments (ratings) of the users.

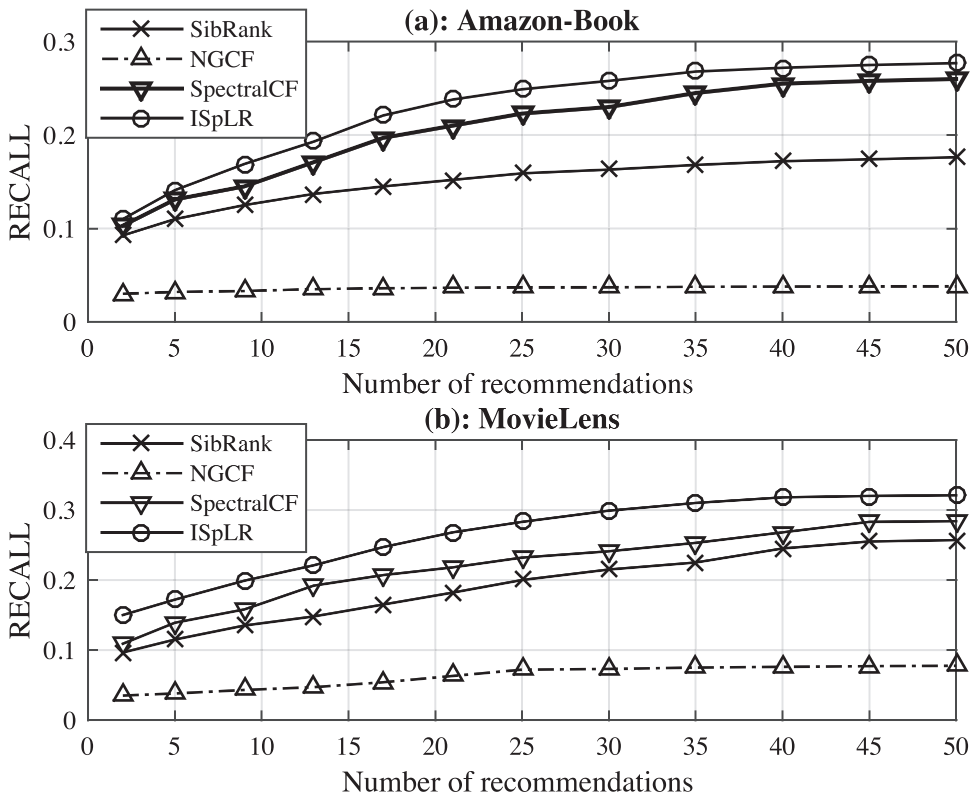

Figure 4. Recovery rate depending on the length of the recommendation list.

The recall metric as shown in Figure 4 with the two sets Amazon-Books (higher sparsity) and MovieLens (lower sparsity). The proposed method (ISpLR) recorded a better performance from a few recommendations (20), the recall rate reached 30%, which represents the rate of relevant items recommended (true positive rate).

NGCF has used several hierarchical layers to aggregate an order of connectivity that lowers the degree of similarity between users and closest items, which negatively affects recall and accuracy.

The last experiment (Figure 5) presents a variation of the proposed methods and focuses on the users as the first starting set. USpLR predicts a link based on the cumulative item-share from the neighbours of the active user as illustrated in the example in the approach section. USpLR (Figure 5) has good results compared to the other methods but lower performance than ISpLR (MAE USpLR=0.709; MAE ISpLR=0.707 for MovieLens). The item approach (ISpLR) is favoured over the user approach (USpLR), knowing that the similarity of the items is closer than the similarity of the user ratings.

For comparison, Table 5 shows some results that compare our approach *(ISpLR)* to some works by mean absolute error.

From these promising results, we can reason that ISpLR offers good quality predictions even in sparse data sets and adapts well to online recommendations. On e-commerce sites, it is possible to retain only a subset of items to generate predictions for the other items.

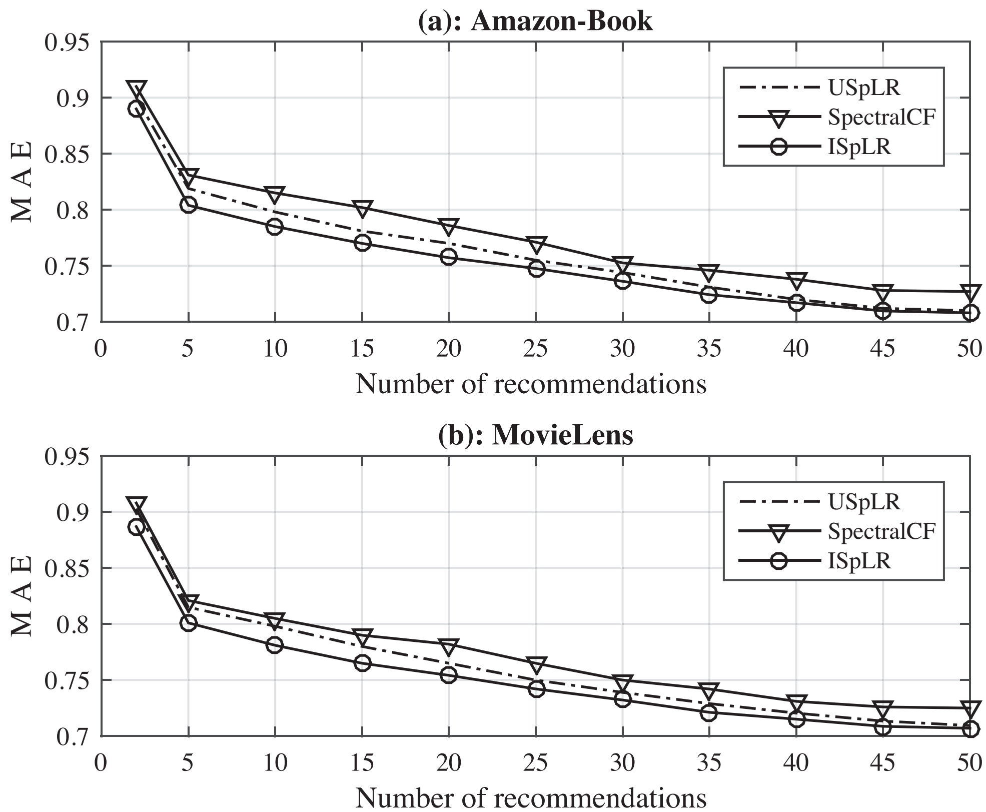

Figure 5. Error of USpLR and ISpLR.

Table 5. Comparison of the prediction error.

Methods	**MAE**
Item based recommendation [1]	0.725
Slope one predictors for online rating [32]	0.743
Incremental SVD [33]	0.768
User-item clustering [34]	0.798
Hybrid semantic recommendation *HSl* [22]	0.874
ISpLR	**0.707**

6. Conclusion and future works

There is a need to model a recommendation system according to a network structure for interpretation and implementation purposes. In this paper, we addressed the problem of link prediction in a weighted bipartite network. According to the weightings of existing links, the weighting of a nonexistent link can be calculated as a predictive value of an item to a user. The idea is to accumulate the shares of each user and each item in a linear and bidirectional way. Share means a part of the sharing of the item towards the user and vice versa. The advantage of this method is twofold first, the accumulation of the shares retains the information propagated between the Items/Users nodes second, the link prediction calculation does not require additional adjustment parameters. Classical collaborative recommendation methods based on Pearson correlation or cosine similarity [1, 3] require several steps to reach the prediction phase. Most link prediction methods [26, 27] focus on neighbour links and apply measures such as the number of neighbours, Katz, Adamic Adar, etc. and do not take the link weight. A link weight measures a user's level of satisfaction with an item on a scale of, say, 1 to 5, which improves recommendation performance. The proposed method offers good results using different dense datasets and minimizes the prediction error compared to competitive methods. In addition, the performance and quality of recommendation are assured through the metrics recall and accuracy. Our future work will be devoted to the scalability problem (combinatorial complexity) of this method compared to existing methods and to see the prediction time for online users.

References

[1] Adomavicius G, Tuzhilinand A. Toward the next generation of recommender systems: a survey of the state-of-theart and possible extensions. IEEE Transactions on Knowledge and Data Engineering 2005; 17 (6): 734-749. doi: 10.1109/TKDE.2005.99

[2] Breese J, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. In: Morgan Kaufmann Publishers Inc The 14th Conference on Uncertainty in Artificial Intelligence; San Francisco, CA, United States; 1998. pp. 43-52.

[3] Schafer BJ, Frankowski D, Herlocker J, Sen S. Collaborative filtering recommender systems. In: Brusilovsky P (editor). Adaptive Web Methods and Strategies of Web Personalization; Springer-Verlag Berlin Heidelberg LNCS (4321), 2007, pp. 291-324.

[4] Sarwar BM, Karypis G, Konstan JA, Riedl J. Item based collaborative filtering recommendation algorithms. In: WWW '01 The 10th International World Wide Web Conference; Hong Kong; 2001. pp. 285–295.

[5] He X, Liao L, Zhang H, Hu X. Neural collaborative filtering. In: ACM Digital Library The 26th International Conference on World Widee Web; Perth Australia; 2017. pp. 173-182.

[6] Ignat'ev YV, Lemtyuzhnikova DV, Rul DI, Ryabov IL. Constructing a Hybrid Recommender System. Journal of Computer and Systems Sciences International 2018; 57 (6): 921-926. doi: 10.1134/S1064230718060060

[7] Musto C, Lops P, Gemmis MD, Semeraro G. Semantics-aware Recommender Systems exploiting Linked Open Data and graph-based features. Knowledge-Based Systems 2017; 136 (C): 1-14. doi: 10.1016/j.knosys.2017.08.015

[8] Yin R, Li K, Zhang G, Lu JA. Deeper graph neural network for recommender systems. Knowledge-Based Systems 2019; 185 (1): doi: 10.1016/j.knosys.2019.105020

[9] Eirinaki M, Gao J, Varlamis I, Tserpes K. Recommender systems for large-scale social networks: A review of challenges and solutions. Future Generation Computer Systems 2018; 78 (1): 413-418. doi: 10.1016/j.future.2017.09.015

[10] He X, Gao M, Kan M. BiRank: Towards ranking on Bipartite Graphs. IEEE Transactions On Knowledge and Data Engineering 2016; 29 (1): 57-71. doi: 10.1109/TDK.2016.2611584

[11] Page L, Brin S, Motwani R, Winograd T. The PageRank citation ranking: Bringing order to the Web. Stanford InfoLab- Digital Libraries, Stanford, USA, 1999.

[12] Liang D, Krishnan RG, Hoffman RG. Variational autoencoders for collaborative filtering. In: WWW'18 International World Wide Web Conferences, Lyon, France; 2018. pp. 689-698.

[13] Elahi M, Ricci F, Rubens N. A survey of active learning in collaborative filtering recommender systems. Computer Science Review 2016; (20): 29-50. doi: 10.1016/j.cosrev.2016.05.002

[14] Ambulgekar HP, Pathak MK, Kokare MB. A survey on collaborative filtering: tasks, approaches and applications. In: International Ethical Hacking Conference; Kolkata, India; 2018. pp. 289-300.

[15] Chen R, Hua Q, Chang YS, Wang B, Zhang L et al. A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks. IEEE Access 2018;(6):64301-64320. doi: 10.1109/ACCESS.2018.2877208

[16] Kovkov DV, Lemtyuzhnikova DV. Decomposition in Multidimensional Boolean-Optimization Problems with Sparse Matrices. Journal of Computer and Systems Sciences International 2018; 57 (1): 97–108. doi: 10.1134/S1064230718010033

[17] Sandhya H, Akhil AS, Ramachandra K. A depth-based nearest neighbor algorithm for high-dimensional data classification. Turkish Journal of Electrical Engineering and Computer Science 2019; 27 (6): 4082-4101. doi: 10.3906/elk-1807-163

[18] Kamer K. Parallel algorithms for computing sparse matrix permanents. Turkish Journal of Electrical Engineering and Computer Science 2019; 27 (6): 4284-4297. doi: 10.3906/elk-1807-163. doi: 10.3906/elk-1904-135

[19] Adomavicius G, Mobasher B, Ricci F, Tuzhilin A. Context Aware Recommender systems. AI Magazine 2011; 32(3): 67-80. doi: 10.1609/aimag.v32i3.2364

[20] Kashevnik AM, Ponomarev AV. A multimodel context-aware tourism recommendation service: Approach and architecture. Journal of Computer and Systems Sciences International 2017; 56 (2): 245–258. doi: 10.1134/S1064230717020125

[21] Navimipour NJ, Keshanchi B, Milani FS. Resources discovery in the cloud environments using collaborative filtering and ontology relations. Electronic Commerce Research and applications 2017; 26: 89-100. doi: 10.1016/j.elerap.2017.10.005

[22] Kharroubi S, Dahmani Y, Nouali O. A semantic layer to improve collaborative filtering systems. International Journal of Computational Science and Engineering 2018; 17 (4): 365-376. doi: 10.1504/IJCSE.2018.096024

[23] Giannoulakis S, Tsapatsoulis S. Filtering Instagram Hashtags through crowdtagging and the HITS algorithm. IEEE Transactions on Computational Social Systems 2019; 6 (3): 592-603. doi: 10.1109/TCSS.2019.2914080

[24] Birkbak A, Carlsen H. The world of Edgerank: Rhetorical justifications of Facebook's News Feed algorithm. Computational Culture 2016; (5).

[25] Nesreen KA, Nick D, Liangzhen X. Estimating Node Similarity by Sampling Streaming Bipartite Graphs. In: Twenty-Seventh International Joint Conference on Artificial Intelligence; Stockholm, Sweden; 2018. pp.3286-3292.

[26] Kim JH, Segev A. Hypothesis Generation Using Link Prediction in a Bipartite Graph. In: IEEE International Conference on Big Data; Seattle, WA, USA; 2018. pp. 2863-2867.

[27] Yiding L, Yufan G, Anna K. Link prediction in drug-target interactions network using similarity indices. BMC Bioinformatics 2017; 18 (39): 1-9. doi: 10.1186/s12859-017-1460-z

[28] Smith B, Linden G. Two decades of recommender systems at Amazon. com. IEEE Internet Computing 2017; 21 (13): 12-18. doi: 10.1109/MIC.2017.72

[29] Shams B, Haratizadeh S. SibRank: Signed bipartite network analysis for neighbor-based collaborative ranking. Physica A: Statistical Mechanics and its Applications 2016; 458: 364-377. doi: 10.1016/j.physa.2016.04.025

[30] Wang X, He X, Wang M, Feng F, Chua TS. Neural graph collaborative filtering. In: ACM SIGIR International Conference on Research and Development in Information Retrieval; Paris, France; 2019. pp.165–174.

[31] Zheng L, Lu CT, Jiang F, Zhang J, Yu PS. Spectral collaborative filtering. In: Twelfth ACM Conference on Recommender Systems; Vancouver British Columbia, Canada; 2018. pp. 311-319.

[32] Lemire D, Maclachlan A. Slope One Predictors for Online Rating-Based Collaborative Filtering. In: SIAM Data Mining; Newport Beach, California; 2005.

[33] Xun Z, Jing H, Guangyan H, Yanchun Z. SVD-based incremental approaches for recommender systems. Journal of Computer and System Sciences 2015; 81 (4): 717-733. doi: 10.1016/j.jcss.2014.11.016

[34] Song JG. A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering 2010; 5 (7): 745-752. doi: 10.4304/jsw.5.7.745-752