The Journal of Supercomputing (2021) 77:6832–6851
https://doi.org/10.1007/s11227-020-03573-8

Exploiting uninteresting items for effective graph-based one-class collaborative filtering

Yeon-Chang Lee[1] · Jiwon Son[1] · Taeho Kim[1] · Daeyoung Park[1] · Sang-Wook Kim[1]

Accepted: 14 December 2020 / Published online: 4 January 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

Abstract

The goal of recommender systems is to identify the items appealing to a target user by analyzing her/his past preferences. Collaborative filtering is one of the most popular recommendation methods that use the similarity between users' past behaviors such as explicit user ratings (i.e., multi-class setting) or implicit click logs (i.e., one-class setting). Graph-theoretic one-class collaborative filtering (gOCCF) has been successful in dealing with sparse datasets in one-class settings (e.g., clicked or bookmarked). In this paper, we point out the problem that gOCCF requires long processing time compared to existing OCCF methods. To overcome the limitation of the original gOCCF, we propose a new gOCCF approach based on signed random walk with restart (SRWR). Using SRWR, the proposed approach accurately and efficiently captures users' preferences by analyzing not only the positive preferences from rated items but also the negative preferences from uninteresting items. We also perform an in-depth analysis to further understand the effect of employing uninteresting items in OCCF. Toward this end, we employ the following well-known graph properties: (1) effective diameter, (2) number of reachable pairs, (3) number of nodes in the largest connected component, (4) clustering coefficient, (5) singular values, and (6) signed butterfly. From this comprehensive analysis, we demonstrate that signed graphs with uninteresting items have properties similar to real-life signed graphs. Lastly, through extensive experiments using real-life datasets, we verify that the proposed approach improves the accuracy and decreases the processing time of the original gOCCF.

Keywords One-class collaborative filtering · Uninteresting items · Graph-based recommendations

✉ Sang-Wook Kim
wook@hanyang.ac.kr

Extended author information available on the last page of the article

1 Introduction

The goal of recommender systems is to identify items appealing to a target user by analyzing their past preferences [33]. Typical examples of items to be recommended are movies [22, 24], music [38, 39], and TV shows [7]. The collaborative filtering (CF) [35] is one of the most popular techniques in recommender systems. CF exploits the past behavior of a user such as explicit feedback represented by their ratings (i.e., a multi-class setting) and implicit feedback represented by his/her clicks (i.e., a one-class setting) to recommend the most appealing top-*N* items to them [35]. Compared to the multi-class setting, the one-class setting has greater availability, thus facing wider applications. To support this type of one-class setting, one-class collaborative filtering (OCCF) methods have been proposed [30, 32, 40].

In our recent work [23], we pointed out the problem that existing OCCF methods tend to become less effective when dealing with sparse datasets. To address this challenge, we proposed a graph-theoretic OCCF approach named as gOCCF that exploits not only the rated items but also the uninteresting items [15, 21] inferred from unrated items. To determine a right number of uninteresting items, it considers the degree of interestingness on unrated items, the graph shattering theory [1], and the property of information propagation. gOCCF has two options that can represent the positive/negative relationships between users and rated/uninteresting items: a single signed graph and two separate graphs having positive and negative links, respectively. It comes up with top-*N* recommendations based on two well-known graph analysis techniques: random walk with restart (RWR) [9] and belief propagation (BP) [16].

Despite the success of the original gOCCF in the sparse one-class setting, we observe that the following two challenges in gOCCF. First, the original gOCCF finds the uninteresting items of all users and also analyzes the relationships between users and uninteresting items along with those between users and rated items. This requires longer processing times than other OCCF methods [5, 30, 32, 40] that exploit only the rated items. Second, SignedBP [16], which is the best performer among the three variants of the original gOCCF, employs an extension of BP to analyze a signed graph. We note that BP has higher model complexity than that of RWR.

In this paper, we propose a graph-theoretic OCCF approach that aims to improve the accuracy and the processing time of the original gOCCF.[1] Toward the two goals, we employ signed random walk with restart (SRWR) [19]. SRWR is a technique that was proposed recently to solve a node ranking problem in a signed graph: It extends the original RWR [9] to account for link signs (i.e., positive and negative), employing a signed surfer to provide more accurate node rankings [19]. In this sense, SRWR is suitable to analyze both positive preferences from rated items and negative

[1] The initial idea of this paper was introduced with some preliminary results of evaluation at IEEE BigComp 2020 as a short paper of four pages (i.e., conference version) [25]. This paper is its extended version written for the archival purpose in a journal.

preferences from uninteresting items. In addition, RWR-based models are known to have lower model complexity than BP-based models.

We perform an in-depth analysis to better understand the effect of employing the uninteresting items in OCCF. First, we compare the topological properties of positive graphs (i.e., relationships between users and their rated items) with those of negative graphs (i.e., relationships between users and their uninteresting items). To do this, we employ five well-known graph properties [1, 18, 20]: (1) effective diameter, (2) number of reachable pairs, (3) number of nodes in the largest connected component, (4) clustering coefficient, and (5) singular values. Through the comparisons, we confirm that positive and negative graphs are similar to each other for most of these properties.

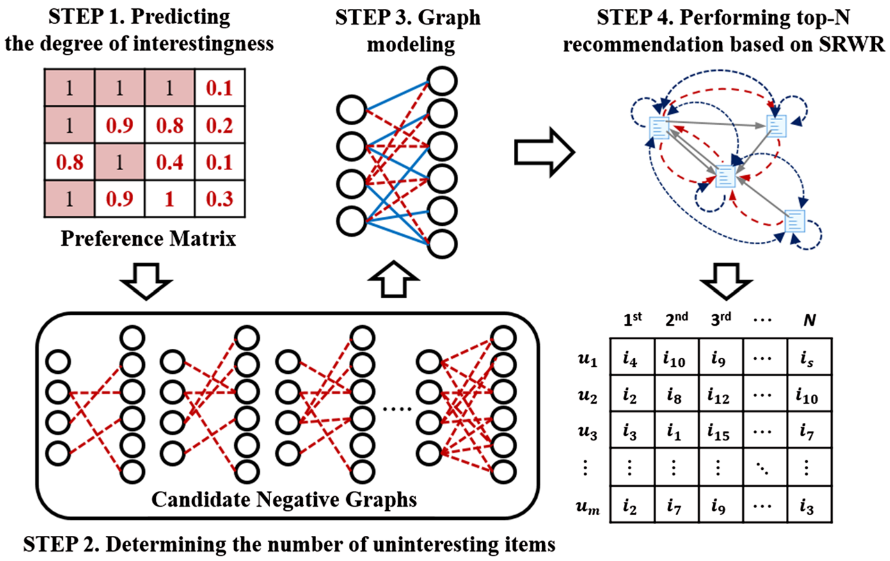
Second, we employ the signed butterfly [8] to analyze the structure of a signed graph modeled with positive/negative relationships. The signed butterfly [8] is a concept proposed to utilize the balance theory [4, 14, 26], which was originally used to explain the social phenomena of a link structure in a signed unipartite graph, in a signed bipartite graph. With this concept, we show that the structure of signed graphs modeled by our approach is similar to that of real-life signed graphs.

Finally, we conduct extensive experiments using real-life datasets to verify the effectiveness and efficiency of our proposed approach. The experimental results show that the proposed approach successfully achieves the above two goals, thereby improving the original gOCCF. Specifically, in the MovieLens 100K dataset, the proposed approach improves the accuracy (in terms of NDCG [17]) of gOCCF by up to 26.87% and reduces the processing time of gOCCF by up to 8.5 times.

Our contributions are summarized as follows:

- We point out the limitations in the performance aspect of the original gOCCF that utilizes the users' rated and uninteresting items together.
- We propose a novel SRWR-based gOCCF that can analyze the rated and uninteresting items of users efficiently.
- We perform an in-depth analysis on the relationships between users and rated/uninteresting items based on the graph theory.
- Via extensive experiments using five real-life datasets, we demonstrate that our proposed approach significantly improves both the accuracy and performance of the original gOCCF.

The rest of this paper is organized as follows: Section 2 reviews the existing OCCF methods. Section 3 presents the proposed approach in detail. Section 4 shows the results of in-depth analysis on signed graphs having uninteresting items. Section 5 validates the effectiveness of the proposed approach through extensive experiments. Finally, Section 6 summarizes and concludes the paper.


STEP 1. Predicting the degree of interestingness
Preference Matrix
STEP 3. Graph modeling
STEP 4. Performing top-N recommendation based on SRWR
Candidate Negative Graphs
....
STEP 2. Determining the number of uninteresting items

1	1	1	0.1
1	0.9	0.8	0.2
0.8	1	0.4	0.1
1	0.9	1	0.3

	1st	2nd	3rd	⋯	N
u_1	i_4	i_{10}	i_9	⋯	i_s
u_2	i_2	i_8	i_{12}	⋯	i_{10}
u_3	i_3	i_1	i_{15}	⋯	i_7
⋮	⋮	⋮	⋮	⋱	⋮
u_m	i_2	i_7	i_9	⋯	i_3

Fig. 1 Overview of the proposed approach

2 Related work

In this section, we briefly review the existing OCCF methods. The traditional OCCF methods have mainly focused on matrix factorization (MF) techniques. They can be classified into the following two categories, according to the way how they learn users' preferences [31]: the point-wise and pair-wise approaches. The point-wise approach aims at minimizing the errors (e.g., root mean square errors) between the real and predicted ratings in predicting the absolute ratings of items given by each user. Typical examples are WRMF [30], eALS [13], and NeuMF [12]. The pair-wise approach aims to maximize the relative preference difference between pairs of the rated and unrated items for each user. Typical examples are BPRMF [32], GBPR [31], and NPR [28].

Recently, various graph-based OCCF methods have been proposed to directly learn collaborative signals of user–item interactions [3, 37]. Among them, neural graph collaborative filtering (NGCF) [37] provides recommendations by analyzing a bipartite graph modeled on the relationship between users and rated items based on graph convolutional network (GCN) [10]. Encouraged by the success of NGCF, many researchers have attempted to exploit GCN for OCCF in recommendation [6, 11, 36].

However, the existing OCCF methods tend to become less effective in dealing with sparse datasets, which are typical in practice. To overcome this problem, our prior work [23] proposed a graph-based OCCF method, named as gOCCF, that utilizes the relationships between users and uninteresting items as well as those between users and rated items. As mentioned in Section 1, however, the original gOCCF requires a long processing time compared to existing OCCF methods.

3 Proposed approach

In this section, we aim at designing a new graph-based OCCF approach that requires a smaller processing time without sacrificing the accuracy of the original gOCCF. As shown in Fig. 1, the proposed approach is composed of the following four steps: (1) predicting the degree of interestingness on unrated items; (2) determining the number of uninteresting items; (3) modeling the relationships between users and rated/uninteresting items; and (4) performing top-N recommendation by employing SRWR.

3.1 Problem definition

We first formulate the top-N recommendation problem as follows. Let $u \in \mathcal{U}$ and $i \in \mathcal{I}$ denote a user and an item, respectively, where $\mathcal{U}$ and $\mathcal{I}$ denote the sets of all users and all items, respectively. Additionally, $\mathcal{I}_{\mathrm{u}}$ denotes a set of items *rated* by user u. For each user u, the goal is to recommend the top-N items that are most likely to be preferred by u among her unrated items (i.e., $\mathcal{I} \setminus \mathcal{I}_{\mathrm{u}}$).

3.2 Overall procedure

Step 1: Predicting the degree of interestingness. We first predict the degree of interestingness on every unrated item for a user by referring to his/her rated items only. Toward this end, we can exploit any existing OCCF methods [5, 30, 32, 40]; in this paper, we employ a popular OCCF method, weighted regularized matrix factorization (WRMF) [30], by following [15, 23]. WRMF represents rated and unrated items as positive (i.e., 1) and negative (i.e., 0) preferences, respectively, in a rating matrix E. It assigns different weights to those preferences for matrix factorization by quantifying their relative confidence. Then, it factorizes E into two low-rank matrices X and Y, each of which represents the features of users or items as latent factors. This matrix factorization employs the weight alternating least square(wALS) technique [34], which reflects the weights assigned to preferences as their relative contributions.

Specifically, wALS first assigns random values to elements in the features of the features of items Y and updates elements in the features of users X as in (Eq. 1) by optimizing the loss function:

$$X_{\mathrm{u}(\cdot)} = E_{\mathrm{u}(\cdot)} \widetilde{C}_{\mathrm{u}(\cdot)} Y \left\{ Y^T \widetilde{C}_{\mathrm{u}(\cdot)} Y + \lambda \left(\sum_s c_{\mathrm{u,s}} \right) I \right\}^{-1}, \tag{1}$$

where u indicates a user, $\widetilde{C}_{\mathrm{u}(\cdot)}$ is a diagonal matrix with the elements of $C_{\mathrm{u}(\cdot)}$ on the diagonal, and I is an identity matrix. After that, wALS updates elements in Y while fixing X as in (Eq. 2):

$$Y_{\mathrm{s}(\cdot)} = E^{T}_{(\cdot)\mathrm{s}} \widetilde{C}_{(\cdot)\mathrm{s}} X \left\{ X^{T} \widetilde{C}_{(\cdot)\mathrm{s}} X + \lambda \left(\sum_{u} c_{\mathrm{u,s}} \right) I \right\}^{-1}. \tag{2}$$

where s indicates a item. Finally, it predicts a user's preference on an unrated item of hers by taking the inner product of two latent factors of the user and the item.

Step 2: Determining the number of uninteresting items. The goal of STEP 2 is to determine a right number of uninteresting items based on graph properties without a parameter θ, unlike zero injection [15] that depends on θ. Based on the degree of interestingness computed by WRMF, we can consider unrated items that have a *low* degree of interestingness as uninteresting items. Note, however, that it is a very time-consuming task to find a right number of uninteresting items that provides the best accuracy in recommendation. For example, in the MovieLens 100K, we need to determine uninteresting items from 1.5M unrated items of users (i.e., user–item pairs).

Following [23], we decide this number by employing the following two graph-theoretic tools: (1) graph shattering theory [1], which introduces a shattering point at which the connectivity of a graph becomes seriously collapsed when links are continuously removed in a random way; and (2) PageRank [29] that is known to best characterize the aspect of information propagation. By analyzing datasets with the above tools, as described in [23], we demonstrated that setting the number of uninteresting items as *equal to* that of rated items helps to get topological properties most similar to those of a given dataset. Furthermore, we showed that this setting provides the highest accuracy in terms of recommendation. Therefore, we determine that there is the same number of uninteresting items as rated items in a rating matrix.

Step 3: Modeling the relationships between users and items. The goal of STEP 3 is to analyze both the positive and negative preferences of users, unlike existing OCCF methods [12, 13, 28, 30–32] that only analyze the positive preferences. Note that we can convert the original dataset in one-class setting to one in the binary-class setting, which has both positive preferences from rated items and negative preferences from uninteresting items. In order to model such binary-class information in graphs, we can consider the following two candidates [23]: (1) two separate graphs modeled by independently considering positive and negative links and (2) a single signed graph modeled by taking both positive and negative links together into account.

We can capture more accurate users' preferences by combining two kinds of relationships: one with users and his/her rated items (i.e., positive links) and the other with users and his/her uninteresting items (i.e., negative links). The first candidate above exploits the information provided by two separate graphs independently. In this paper, we model the binary-class information in a single signed graph. In the signed graph, the nodes represent users and items, the positive links represent the relationships between users and their rated items, and the negative links represent those between users and their uninteresting items.

Step 4: Performing top-N recommendation. The goal of STEP 4 is to provide high accuracy with a small processing time, unlike the original gOCCF [23] that requires a large processing time. To provide an efficient and effective recommendation, we employ the signed random walk with restart (SRWR) technique [19]. To do this, we first define the following two probabilities: (1) $r^+_{\mathrm{u}} = \mathrm{Pr}_{\mathrm{t}}(u, +)$ is the probability that the positive surfer arrives at node (i.e., user or item) u after SRWR from the target user t and (2) $r^-_{\mathrm{u}} = \mathrm{Pr}_{\mathrm{t}}(u, -)$ is the probability that the negative surfer arrives at node (i.e., user or item) u after SRWR from the target user t. r^+ indicates a positive SRWR score vector and r^- is a negative SRWR score vector for all nodes. Both vectors are used for predicting the preferences on all unrated items of a target user t.

To update two vectors r^+ and r^-, we formulate the following recursive equations [19]:

$$\begin{aligned} r^+ &= (1-c)(\tilde{A}_+^\top r^+ + \beta\tilde{A}_-^\top r^- + (1-\gamma)\tilde{A}_+^\top r^-) + \mathrm{cq}, \\ r^- &= (1-c)(\tilde{A}_-^\top r^+ + \gamma\tilde{A}_+^\top r^- + (1-\beta)\tilde{A}_-^\top r^-), \end{aligned} \tag{3}$$

where c is a restarting probability, β is a parameter for controlling the uncertainty of "the enemy of my enemy is my friend," and γ is a parameter for controlling the uncertainty of "the friend of my enemy is my enemy"; q is a vector whose tth element (i.e., target user) is set as 1 and all other elements are set as 0; $\tilde{A}_+$ and $\tilde{A}_-$ are the positive and negative semi-row normalized matrixes, respectively.

Algorithm 1 SRWR-based gOCCF

Input: a set of users $\mathcal{U}$, a set of items $\mathcal{I}$, restart probability c, balance attenuation factors β, and γ, and an adjacency matrix A
Output: r^+ and r^-

```
Predict the degree of interestingness for unrated items of all users
for each user $u \in \mathcal{U}$ do
    Divide $\mathcal{I}$ into ${\mathcal{I}_u}^{rt}$ and ${\mathcal{I}_u}^{uni}$ for $u$
    $\widetilde{A}_{u,i_u^{rt}} \leftarrow 1$, $i_u^{rt} \in {\mathcal{I}_u}^{rt}$
    $\widetilde{A}_{u,i_u^{uni}} \leftarrow -1$, $i_u^{uni} \in {\mathcal{I}_u}^{uni}$
end for
Compute outdegree matrix $D$ and semirow normalized matrix $\widetilde{A}$
Split $\widetilde{A}$ into $\widetilde{A}_+$ and $\widetilde{A}_-$ such that $\widetilde{A} = \widetilde{A}_+ - \widetilde{A}_-$
for each user $u \in \mathcal{U}$ do
    Set the starting vector $q$ from $u$
    Set $r^+ = q$, $r^- = 0$, and $r' = [r^+; r^-]^T$
    for Converge do
        Compute $r^+$ and $r^-$ using (Eq. 3)
        Concatenate $r^+$ and $r^-$ into $r = [r^+; r^-]^T$
        Update $r' \leftarrow r$
    end for
end for
```

The iteration steps of SRWR are summarized as follows. First, we compute r^+ and r^- via (Eq. 3) and concatenate r^+ and r^- into r. Then, we compute the error between r and r' where r' is the result obtained in the previous iteration. We set r as r' for the next iteration. The iteration is repeated in the same way until the error

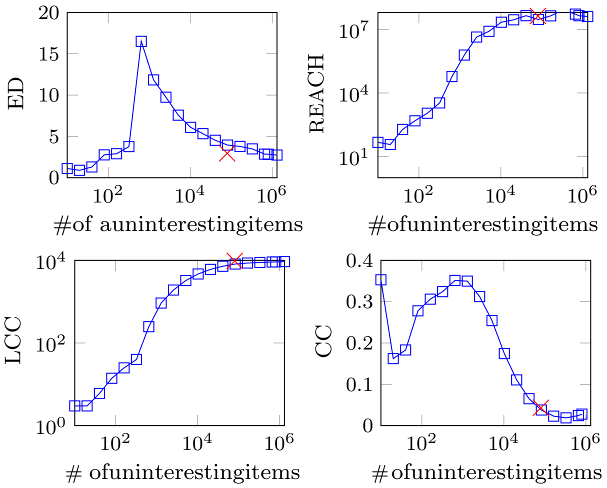


Fig. 2 Topological properties for a real positive graph and its negative graphs modeled by our approach

becomes smaller than a threshold predefined. Finally, we recommend the top-N items that are most preferred by the target user based on the final SRWR scores computed by $r^+ - r^-$, both contained in r.

Algorithm 1 sketches the overall process of our proposed approach. We first predict the degree of interestingness for unrated items of all users (line 1). For each user u, we divide $\mathcal{I}_\mathrm{u}$ into $\mathcal{I}_\mathrm{u}^\mathrm{rt}$ and $\mathcal{I}_\mathrm{u}^\mathrm{uni}$ and fill the values of A based on them (lines 2–6). Next, we compute D and $\widetilde{A}$, and then split$\widetilde{A}$ into $\widetilde{A}_+$ and $\widetilde{A}_-$. (lines 7–8). Now, we are ready to perform SRWR per user u (line 9). First, we set q to uth unit vector and initialize r^+ and r^- (lines 10–11). Then, we compute the r^+ and r^- for u and concatenate r^+ and r^- vertically (lines 13–14). Finally, we update r into r' for the next iteration (line 15). The time complexity of our approach is $O(T * (|\mathcal{U}| + |\mathcal{I}| + r))$, where T indicates the number of iterations in SRWR, $|\mathcal{U}|$ and $|\mathcal{I}|$ do the numbers of users and items, respectively, and r indicates the number of ratings.

4 Analysis on uninteresting items

In this section, we present an in-depth analysis on the relationships between users and uninteresting items inferred by the proposed approach. Toward this end, we employ the graph shattering theory [1] that introduces a "shattering point" at which the connectivity of a graph becomes seriously collapsed as links are continuously removed in a random way. ShatterPlot is a tool for visualizing the process of generating the shattering point as a plot. By removing links of a graph continuously, this plot helps identify the changes in the topological properties of a graph.

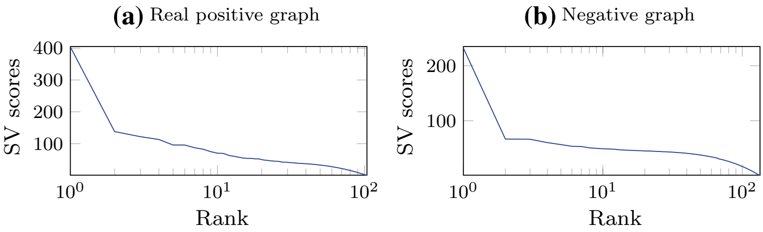


Fig. 3 SV distributions for a real positive graph and its negative graph modeled by our approach

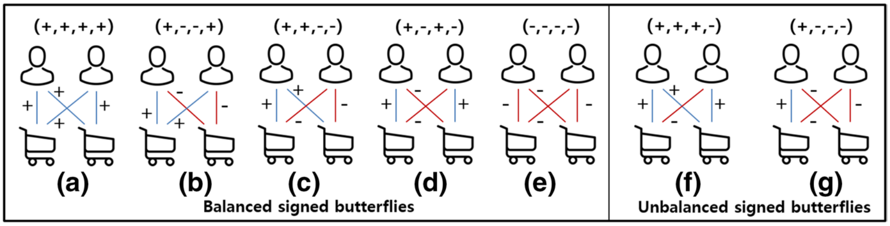


Fig. 4 Seven types of signed butterflies in a signed bipartite graph

In the same way as in [1], we analyze the generation process of the relationships between users and uninteresting items. To do this, we consider k unrated user–item pairs with the lowest degree of interestingness and model them to have negative links in a single negative graph. In this case, starting from 10, we double k until the negative links reach 90% of the unrated user–item pairs. Then, we compare the topological properties [1, 20] of a real positive graph with those of negative graphs modeled by our approach. Analyzed properties include the following: (1) effective diameter (ED), (2) number of reachable pairs (REACH), (3) number of nodes in the largest connected component (LCC), and (4) clustering coefficient (CC).

Figure 2 shows the changes in topological properties of our negative graphs in MovieLens 1M dataset.[2] The x-axis represents the number of links in a graph, and the y-axis represents the value of each topological property in the graph. The mark "x" indicates the topological property in the real positive graph. We see that, for all the topological properties, when the negative graph has the same number of links as the real positive graph, it shows values most similar to those of its real positive graph. This trend coincides with those reported in [23].

Now, we examine the negative graph that has the same number of links as that of the positive graph in more detail. To this end, we compare the negative graph with the positive graph based on the singular values of adjacency matrix (SV) [20]. Figure 3 shows the SV distributions of the positive and negative graphs in the MovieLens 1M dataset.[3] The x-axis represents top-k ranks, and y-axis represents an SV

[2] Note that the same trend was observed as well in other datasets (i.e., MovieLens 100K, Watcha, CiteULike, and Lastfm).

[3] The same trend was also observed in other datasets (i.e., MovieLens 100K, Watcha, CiteULike, and Lastfm).

Table 1 Statistics on signed butterflies on the signed graph modeled by our approach

(a) MovieLens 100K

Types	Balanced					Unbalanced	
	(a)	(b)	(c)	(d)	(e)	(f)	(g)
Ratio	43%	12%	11%	7%	6%	15%	6%
	79%					21%	

(b) MovieLens 1M

Types	Balanced					Unbalanced	
	(a)	(b)	(c)	(d)	(e)	(f)	(g)
Ratio	51%	10%	11%	4%	6%	14%	4%
	82%					18%	

(c) Watcha

Types	Balanced					Unbalanced	
	(a)	(b)	(c)	(d)	(e)	(f)	(g)
Ratio	26%	14%	13%	12%	10%	15%	10%
	75%					25%	

(d) CiteULike

Types	Balanced					Unbalanced	
	(a)	(b)	(c)	(d)	(e)	(f)	(g)
Ratio	19%	10%	7%	9%	43%	2%	10%
	88%					12%	

(e) Lastfm

Types	Balanced					Unbalanced	
	(a)	(b)	(c)	(d)	(e)	(f)	(g)
Ratio	27%	9%	15%	7%	26%	6%	10%
	84%					16%	

score for each rank. From Fig. 3, we confirm that the negative graph whose number of links is the same as that of the positive graph follows the power-law distribution as in the positive graph.

Finally, we model the positive and negative graphs as a single signed graph and analyze the structure of the signed graph. To this end, we employ the signed butterfly [8]. This concept was proposed to utilize the balance theory [4, 14, 26], which was originally used to explain the social phenomena of a link structure in a signed unipartite graph, in a signed bipartite graph. As shown in Fig. 4, Derr et al. [8] defined seven types of signed butterflies to represent the relationships among two users and two items in a signed bipartite graph. Figure 4a–e represents balanced signed butterflies, while Fig. 4f and g shows unbalanced signed butterflies. In [8],

Table 2 Statistics of datasets

Datasets	# Users	# Items	# Feedback	Sparsity (%)
Last. fm	1862	17,631	92,834	99.8
ML100K	943	1682	100,000	93.7
Watcha	1391	1927	100,000	96.9
CiteULike	5551	16,980	210,504	99.8
ML1M	6040	3,900	1,000,209	95.7

the authors showed that, similar to signed unipartite graphs, most real-life signed bipartite graphs have many more balanced signed butterflies than unbalanced signed butterflies. For example, in the US Senate dataset, the proportions of balanced and unbalanced signed butterflies were 79% and 21%, respectively [8].

Now, we analyze the signed butterfly of a signed graph modeled by our approach. Table 1 shows the results for our five datasets, i.e., MovieLens 100K, MovieLens 1M, Watcha, CiteULike, and Lastfm. In all datasets, we see that the ratio for balanced signed butterflies is significantly higher than that for unbalanced signed butterflies. Specifically, in the MovieLens 1M dataset, the ratios of balanced and unbalanced signed butterflies are 82% and 18%, respectively. The results also indicate that the signed graph modeled by our approach has a structure very similar to that of real-life signed graphs.

5 Evaluation

In this section, we show empirical evidences on the effectiveness of the proposed approach via extensive experiments. We attempt to answer the following six research questions (RQs):

- RQ1: Which methods perform the best at inferring the degree of interestingness on unrated items to find uninteresting items?
- RQ2: Do the uninteresting items selected by our decision lead to a more accurate recommendation?
- RQ3: Is it useful to exploit additional uninteresting items with existing graph-based recommendations?
- RQ4: How are the accuracy and performance of our approach in very sparse datasets, compared to existing OCCF methods?
- RQ5: How are the scalability of our approach?
- RQ6: How does the accuracy of our approach change with different values of parameters?

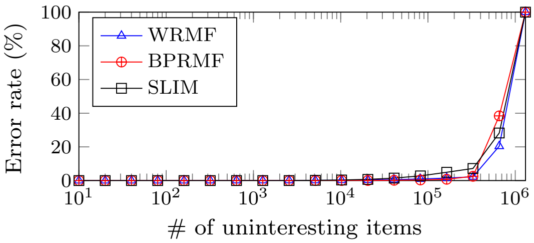


Fig. 5 Error rates according to the number of uninteresting items

5.1 Experimental settings

For evaluation, we used five real-life datasets, MovieLens 100K (ML100K, in short), MovieLens 1M (ML1M, in short), Watcha, CiteULike, and Lastfm. ML100K, ML1M, and Watcha contain ratings ranging from 1 to 5, which are explicitly given by users (i.e., a multi-class setting). Since our goal is to address the OCCF problem, we convert their 1–5 ratings into 1 (i.e., a one-class setting), as popularly done in other OCCF researches [23, 30, 32]. CiteULike and Lastfm contain only implicit feedback and originally use a one-class setting. Table 2 shows the detailed statistics of the datasets.

To evaluate the recommendation accuracy, we performed top-N (=10, 20, 50) recommendation and measured its accuracy in terms of the following three measures, which are popularly used in a recommendation research. First, we use the F1 score that considers precision and recall together:

$$P_{\mathrm{u}}@N = \frac{|\mathrm{Rel}_{\mathrm{u}} \cap N_{\mathrm{u}}|}{|N_{\mathrm{u}}|}, R_{\mathrm{u}}@N = \frac{|\mathrm{Rel}_{\mathrm{u}} \cap N_{\mathrm{u}}|}{|\mathrm{Rel}_{\mathrm{u}}|}, \tag{4}$$

$$F1_{\mathrm{u}}@N = 2 \cdot \frac{P_{\mathrm{u}}@N \cdot R_{\mathrm{u}}@N}{P_{\mathrm{u}}@N + R_{\mathrm{u}}@N}, \tag{5}$$

where N_u denotes a set of N items that each method recommends to u, and Rel_u denotes a set of items that are considered relevant to u (i.e., ground truth). $P_{\mathrm{u}}@N$ and $R_{\mathrm{u}}@N$ denote precision and recall at N, respectively.

Second, we employ the normalized discounted cumulative gain (NDCG) [17] to reflect the importance of ranked positions of items in N_{u}. Let y_{k} represent a binary variable for the kth item i_{k} in N_{u}, i.e., $y_{\mathrm{k}} \in \{0, 1\}$. If $i_k \in \mathrm{Rel}_{\mathrm{u}}$, y_{k} is set to 1. Otherwise, y_{k} is set to 0. In this case, $\mathrm{NDCG}_{\mathrm{u}}@N$ is computed by:

$$\mathrm{NDCG}_{\mathrm{u}}@N = \frac{\mathrm{DCG}_{\mathrm{u}}@N}{\mathrm{IDCG}_{\mathrm{u}}@N}, \mathrm{DCG}_{\mathrm{u}}@N = \sum_{k=1}^{N} \frac{2^{y_k} - 1}{\log_2 (k+1)}, \tag{6}$$

where $\mathrm{IDCG}_{\mathrm{u}}@N$ is the ideal DCG at N, i.e., for every item i_{k} in N_{u}, y_{k} is set to 1.

Lastly, we employ the half-life utility (HLU) [2] that assumes a user will prefer each consecutive item in the list with an exponential decay of possibility. $\mathrm{HLU}_{\mathrm{u}}$ is computed by:

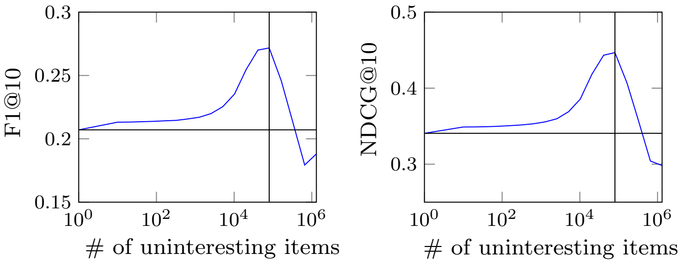


Fig. 6 Accuracy of the proposed approach according to the number of uninteresting items

Table 3 Error rate and accuracy of the proposed approach with different selection methods for uninteresting items

Methods	Error rate (%)	F@10	NDCG@10	HLU
Top U-items	0.86	0.272	0.447	55.10
Top I-items	5.06	0.114	0.285	37.214
Random	55.20	0.166	0.376	45.328

$$\mathrm{HLU_u} = \sum_{k=1} \frac{\delta(k)}{2^{(k-1)(\beta-1)}}, \tag{7}$$

where $\delta(k)$ equals 1 if $i_k \in \mathrm{Rel_u}$ and 0 otherwise, β is the half-life parameter, which is set to 5 in this paper.

The final accuracy for each measures was determined by averaging the five accuracies obtained from five-fold cross-validation. All experiments were conducted on a PC with Intel Core i7 processor (3.70 GHZ) and 64GB RAM.

5.2 Experimental results

RQ1. Comparisons of methods for inferring the degree of interestingness. We first perform a comparative experiment to find the best method that can be used to infer the degree of interestingness of items in Step 1 of our approach. We employ a measure called the error rate [15] that captures how many interesting items are misclassified as uninteresting items per user by an inference method. A user's overall error rate is thus calculated as: err = $\sum_i \frac{|I_i^{U\text{-items}} \cap I_i^{\text{test}}|}{|I_i^{\text{test}}|}$ where $I_\mathrm{i}^{\text{test}}$ is a set of items rated by user i in a test set and $I_\mathrm{i}^{U\text{-items}}$ is a set of items determined as uninteresting items of the user i by a particular inference method. The lower the user's error rate, the better the accuracy of the inference method.

Table 4 Accuracy of the origianl gOCCF and the proposed approach

Methods	gOCCF			Ours
	SeparateRWR	SeparateBP	SignedBP	
(a) ML100K				
F1@10	0.216	0.212	0.263	**0.272**
F1@20	0.260	0.252	0.302	**0.314**
F1@50	0.261	0.259	0.292	**0.306**
NDCG@10	0.352	0.365	0.438	**0.447**
NDCG@20	0.354	0.356	0.426	**0.437**
NDCG@50	0.387	0.385	0.451	**0.466**
HLU	43.89	47.19	**55.98**	55.10
(b) Watcha				
F1@10	0.109	0.117	**0.146**	**0.146**
F1@20	0.128	0.137	**0.164**	**0.164**
F1@50	0.126	0.128	0.147	**0.150**
NDCG@10	0.135	0.151	0.187	**0.188**
NDCG@20	0.158	0.173	**0.212**	**0.212**
NDCG@50	0.217	0.227	0.272	**0.275**
HLU	14.28	17.44	21.96	**22.62**
(c) Lastfm				
F1@10	0.160	0.138	0.155	**0.167**
F1@20	0.158	0.132	0.144	**0.161**
F1@50	0.119	0.102	0.111	**0.121**
NDCG@10	0.192	0.179	0.203	**0.214**
NDCG@20	0.216	0.201	0.223	**0.242**
NDCG@50	0.267	0.248	0.274	**0.295**
HLU	33.13	29.14	**33.77**	33.72

The bold value indicates the best accuracy in each row

Figure 5 shows the changes in the error rates of three OCCF methods, i.e., WRMF [30], BPRMF [32], and SLIM [27], on ML100K. The x-axis represents the number of uninteresting items, and the y-axis represents the error rate measured by each method. In general, as the number of uninteresting items increases, the error rates of all methods increase as well. In addition, the error rates of WRMF are always the lowest, regardless of the number of uninteresting items. Thus, we employed WRMF in the following experiments.

RQ2. Effectiveness of our decision method for uninteresting items. Note that we decided to select the same number of uninteresting items (having the lowest degree of interestingness) as the number of rated items in the given dataset. We first conducted an experiment to verify whether our decision really helps improve the recommendation accuracy. Figure 6 shows top-10 recommendation accuracies of our proposed approach according to different numbers of uninteresting items in ML100K.[4]

[4] We omit the results for other top-Ns (=20, 50), because they showed tendency very similar to that in Fig. 6.

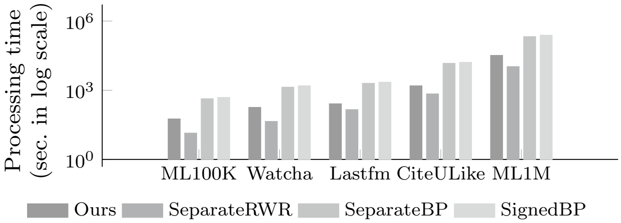


Fig. 7 Processing time of the original gOCCF and the proposed approach

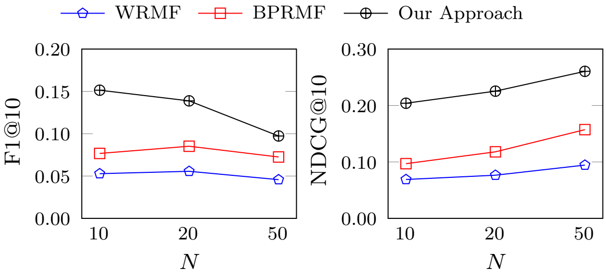


Fig. 8 Accuracy of our approach and competing methods

Table 5 Processing times of our approach and competing methods

Methods	WRMF	BPRMF	Our approach
Processing times (sec.)	603.95	944.10	1539.54

The *x*-axis represents the number of uninteresting items, and the *y*-axis represents the recommendation accuracy measured. The vertical line shows the number of rated items, and the horizontal line shows the accuracy of the original RWR without using the concept of uninteresting items.

The results show that the proposed approach always provides the highest accuracy when the number of uninteresting items is the same as the number of rated items. This trend coincides with those reported in [23]. Also, we found that exploiting uninteresting items is quite effective in improving the accuracy of the original RWR. Figure 6 shows that our approach, where the number of uninteresting items is the same as that of rated items, consistently and significantly outperforms the original RWR. More specifically, our approach improves NDCG@10 of the original RWR by 31.17%. As a result, in subsequent experiments, we set the number of uninteresting items equal to the number of rated items.

Furthermore, in order to verify whether it is appropriate to select negative links based on the degree of interestingness, we compared the links selected by the following three methods: (1) ascending order of the degree of interestingness (i.e., top U-items), (2) descending order of the degree of interestingness (i.e., top I-items), and (3) random. Table 3 shows the error rates of uninteresting items selected by

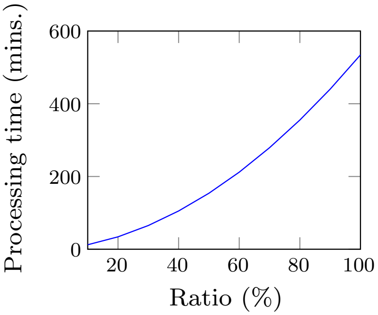


Fig. 9 Processing times with increasing number of ratings

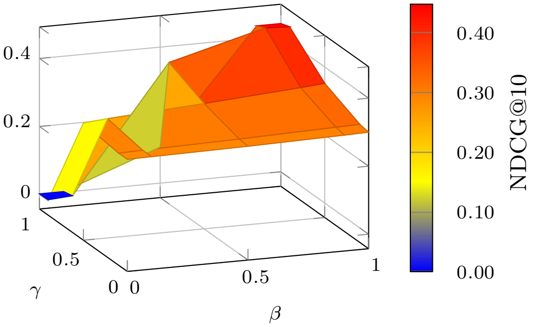


Fig. 10 Accuracy changes according to ***β*** and ***γ***

these methods and the recommended accuracies of our approach equipped with them in ML100K. The result shows the case with the top U-items always provides the lowest error rate and the highest accuracy, followed by the random and the top I-items. We observe that it is the most useful to use the low degree of interestingness as a criterion for selecting links.

RQ3. Accuracy of graph-based recommendation exploiting uninteresting items. We performed an experiment to compare the recommendation accuracies of the original gOCCF with those of our approach. Table 4 shows the accuracies for three variants of the original gOCCF (i.e., SeparateRWR, SeparateBP, and SignedBP) and our approach. The value in boldface indicates the best accuracy in each row. The results show that our approach not only achieves significantly higher accuracy than those of SeparateRWR and SeparateBP, but it also provides accuracy values that are higher than or comparable to that of SignedBP. Specifically, our approach improves NDCG@10 of the original gOCCF (i.e., SeparateBP) by up to 26.87%, 39.60%, and 19.55% for ML100K, Watcha, and Lastfm, respectively.

Moreover, we evaluated the performance of our approach by comparing it with the original gOCCF. Figure 7 shows the processing time of SeparateRWR, SeparateBP, SignedBP, and our approach. The *x*-axis represents the methods in each dataset, and the *y*-axis does the processing time. We summarize the results shown in Fig. 7 as follows. First, both the proposed approach and SeparateRWR require low processing times in all datasets. However, we note that our proposed approach significantly outperforms SeparateRWR in terms of recommendation accuracy (see Table 4). Second, the proposed approach always provides higher performance than SeparateBP and SignedBP. Specifically, for ML100K, the proposed approach reduces the processing time of SeparateBP and SignedBP by 7.5 times and 8.5 times, respectively.

RQ4. Comparison with the existing OCCF methods. We conducted a series of experiments that evaluate and compare the accuracy and performance of our approach and two existing OCCF methods: WRMF and BPRMF. For accuracy comparison, we evaluated the accuracies of the proposed approach, WRMF [30] and BPRMF [32] in the CiteULike, a real-life one-class setting dataset, the most sparse one among the five datasets used in evaluation. The results in Fig. 8 show that our approach consistently outperforms the two competing methods in all metrics and in all top-Ns. Specifically, it dramatically improves NDCG@10 of WRMF and BPRMF by 110.43% and 196.02%, respectively.

For performance comparison, we measured the processing times of our approach, WRMF and BPRMF. Table 5 shows the processing times of our approach and competing methods in CiteULike. We observe that our approach requires the processing time longer than those of other competing methods. However, considering its significant improvement in recommendation accuracy over competing methods (see Fig. 8), this longer execution time can be considered still worth.

RQ5. Scalability of the proposed approach. We examine the change of the processing times of our approach as the number of ratings increases. Toward this end, we measured the processing time of our approach by increasing the ratio of ratings used in execution to all the original ratings in ML1M, the largest dataset, from 10% to 100%. Figure 9 shows the result where the x-axis represents the ratio of ratings and the y-axis represents the processing time. The results demonstrate that our approach has almost a linear scalability with the increasing number of ratings, which indicates our approach is scalable in training with regard to the number of ratings in a dataset.

RQ6. Parameter analysis for our approach. We carefully analyzed the changes of recommendation accuracy of our approach according to different values of parameters: (1) β is a parameter for controlling the uncertainty of “the enemy of my enemy is my friend” and (2) γ is a parameter for controlling the uncertainty of “the friend of my enemy is my enemy.”

Figure 10 shows the changes of accuracy in terms of NDCG@10 by our proposed approach with different values of β and γ in ML100K.[5] The x-axis, y-axis, and z-axis in both figures represent the value of β, the value of γ, and the accuracy, respectively. Figure 10 shows that the larger values of β and γ help achieve high accuracy in all datasets. In particular, the results show that the accuracy is always the highest when β and γ are set as 0.9. These results coincide with those reported in [19],

[5] We omit the results for other top-Ns (=20, 50) because they showed tendency similar to that in Fig. 10.

indicating that considering the uncertainty of the friendship of an enemy is beneficial to improving the recommendation accuracy.

6 Conclusions

In this paper, we proposed a new graph-theoretic OCCF approach based on SRWR. The proposed approach is composed of the following four steps: (1) predicting the degree of interestingness on unrated items; (2) determining uninteresting items based on users' interestingness, the graph shattering theory, and information propagation; (3) modeling the relationships between users and rated/uninteresting items in a signed graph; and (4) performing top-*N* recommendation by employing SRWR on the graph. We performed an in-depth analysis on the relationships between users and uninteresting items inferred by the proposed approach, in terms of the following well-known graph properties: (1) effective diameter, (2) number of reachable pairs, (3) number of nodes in the largest connected component, (4) clustering coefficient, (5) singular values, and (6) signed butterfly. From this comprehensive analysis, we confirmed that the signed graph modeled by our approach has the properties similar to the real-life signed graphs. Through extensive experiments, we also showed that our approach significantly improves the accuracy of gOCCF (i.e., a graph-theoretic state-of-the-art OCCF) and requires less processing time than gOCCF. Specifically, for ML 100K, our approach improves NDCG of gOCCF by up to 26.87% and reduces the processing time of gOCCF by up to 8.5 time.

Acknowledgements This work was supported by Institute of Information and Communications Technology Planning and Evaluation(IITP) Grant funded by the Korea government(MSIT) (No. 2020-0-01373, Artificial Intelligence Graduate School Program (Hanyang University)) and the National Research Foundation of Korea(NRF) Grant funded by the Korea government(MSIT) (No. NRF-2020R1A2B5B03001960).

References

1. Appel A, Chakrabarti D, Faloutsos C, Kumar R, Leskovec J, Tomkins A (2009) ShatterPlots: Fast tools for mining large graphs. In: Proceedings of the SIAM International Conference on Data Mining (SDM), pp 802–813
2. Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI), pp 43–52
3. Cai X, Shang J, Jin Z, Liu F, Qiang B, Xie W, Zhao L (2020) DBGE: employee turnover prediction based on dynamic bipartite graph embedding. IEEE Access 8:10390–10402
4. Cartwright D, Harary F (1956) Structural balance: a generalization of heider's theory. Psycholo Rev 63(5):277
5. Chae D, Kang J, Kim S, Lee J (2018) CFGAN: A generic collaborative filtering framework based on generative adversarial networks. In: Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM), pp 137–146
6. Chen L, Wu L, Hong R, Zhang K, Wang M (2020) Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), pp 27–34

7. Cho K, Lee Y, Han K, Choi J, Kim S (2019) No, that's not my feedback: TV show recommendation using watchable interval. In: Proceedings of IEEE International Conference on Data Engineering (ICDE), pp 316–327
8. Derr T, Johnson C, Chang Y, Tang J (2019) Balance in signed bipartite networks. In: Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM), pp 1221–1230
9. Fouss F, Pirotte A, Renders JM, Saerens M (2007) Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Trans Know Data Eng 19(3):355–369
10. Hamilton WL, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. In: Annual Conference on Neural Information Processing Systems (NIPS), pp 1024–1034
11. He X, Deng K, Wang X, Li Y, Zhang Y, Wang M (2020) Lightgcn: Simplifying and powering graph convolution network for recommendation. In: Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR), pp 639–648
12. He X, Liao L, Zhang H, Nie L, Hu X, Chua T (2017) Neural collaborative filtering. In: Proceedings of The Web Conference (WWW), pp 173–182
13. He X, Zhang H, Kan M, Chua T (2016) Fast matrix factorization for online recommendation with implicit feedback. In: Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR), pp 549–558
14. Heider F (1946) Attitudes and cognitive organization. J Ppsychol 21(1):107–112
15. Hwang WS, Parc J, Kim SW, Lee J, Lee D (2016) "Told You I Didn't Like It": Exploiting uninteresting items for effective collaborative filtering. In: Proceedings of the IEEE International Conference on Data Engineering (ICDE), pp 349–360
16. Jang MH, Faloutsos C, Kim SW, Kang U, Ha J (2016) PIN-TRUST: Fast trust propagation exploiting positive, implicit, and negative information. In: Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM), pp 629–638
17. Järvelin K, Kekäläinen J (2000) IR evaluation methods for retrieving highly relevant documents. In: Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR), pp 41–48
18. Jo Y, Jang M, Kim S, Park S (2019) Realgraph: A graph engine leveraging the power-law distribution of real-world graphs. In: Proceedings of The Web Conference (WWW), pp 807–817
19. Jung J, Jin W, Lee S, Kang U (2016) Personalized ranking in signed networks using signed random walk with restart. In: Proceedings of the IEEE International Conference on Data Mining (ICDM), pp 973–978
20. Jung J, Park H, Kang U (2020) BalanSiNG: Fast and scalable generation of realistic signed networks. In: Proceedings of the International Conference on Extending Database Technology (EDBT), pp 193–204
21. Lee J, Hwang W, Parc J, Lee Y, Kim S, Lee D (2019) l-injection: toward effective collaborative filtering using uninteresting items. IEEE Trans Knowl Data Eng 31(1):3–16
22. Lee J, Lee D, Lee Y, Hwang W, Kim S (2016) Improving the accuracy of top-n recommendation using a preference model. Inf Sci 348:290–304
23. Lee Y, Kim S, Lee D (2018) gOCCF: Graph-theoretic one-class collaborative filtering based on uninteresting items. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), pp 3448–3456
24. Lee Y, Kim S, Park S, Xie X (2018) How to impute missing ratings?: Claims, solution, and its application to collaborative filtering. In: Proceedings of The Web Conference (WWW), pp 783–792
25. Lee Y, Park D, Son J, Kim T, Kim S (2020) Graph-theoretic one-class collaborative filtering using signed random walk with restart. In: Proceedings of the IEEE International Conference on Big Data and Smart Computing (BigComp), pp 98–101
26. Lee Y, Seo N, Han K, Kim S (2020) Asine: Adversarial signed network embedding. In: Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR), pp 609–618

27. Ning X, Karypis G (2011) SLIM: sparse linear methods for top-n recommender systems. In: Cook DJ, Pei J, Wang W, Zaïane OR, Wu X (eds) Proceedings of the IEEE International Conference on Data Mining (ICDM), pp 497–506
28. Niu W, Caverlee J, Lu H (2018) Neural personalized ranking for image recommendation. In: Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM), pp 423–431
29. Page L, Brin S (1999) The PageRank citation ranking: bringing order to the web. Stanford InfoLab, California
30. Pan R, Zhou Y, Cao B, Liu N, Lukose R, Scholz M, Yang Q (2008) One-class collaborative filtering. In: Proceedings of the IEEE International Conference on Data Mining (ICDM), pp 502–511
31. Pan W, Chen L (2013) GBPR: group preference based bayesian personalized ranking for one-class collaborative filtering. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pp 2691–2697
32. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2009) BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI), pp 452–461
33. Ricci F, Rokach L, Shapira B (eds) (2015) Recommender systems handbook. Springer, NY
34. Srebro N, Jaakkola T (2003) Weighted low-rank approximations. In: Proceedings of the International Conference on Machine Learning (ICML), pp 720–727
35. Su X, Khoshgoftaar TM (2009) A survey of collaborative filtering techniques. Adv Artif Intell 2009:421425:1–421425:19
36. Sun J, Guo W, Zhang D, Zhang Y, Regol F, Hu Y, Guo H, Tang R, Yuan H, He X, Coates M (2020) A framework for recommending accurate and diverse items using bayesian graph convolutional neural networks. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pp 2030–2039
37. Wang X, He X, Wang M, Feng F, Chua T (2019) Neural graph collaborative filtering. In: Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR), pp 165–174
38. Wang X, Rosenblum DS, Wang Y (2012) Context-aware mobile music recommendation for daily activities. In: Proceedings of the ACM International Conference on Multimedia Conference (MM), pp 99–108
39. Wang X, Wang Y (2014) Improving content-based and hybrid music recommendation using deep learning. In: Proceedings of the ACM International Conference on Multimedia (MM), pp 627–636
40. Wu G, Volkovs M, Soon CL, Sanner S, Rai H (2019) Noise contrastive estimation for one-class collaborative filtering. In: Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR), pp 135–144

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Yeon-Chang Lee[1] · Jiwon Son[1] · Taeho Kim[1] · Daeyoung Park[1] · Sang-Wook Kim[1]

Yeon-Chang Lee
lyc0324@hanyang.ac.kr

Jiwon Son
tinybeing@hanyang.ac.kr

Taeho Kim
hirooms2@hanyang.ac.kr

Daeyoung Park
mainthread4929@gmail.com

[1] Department of Computer and Software, Hanyang University, Seoul, South Korea