

gOCCF: Graph-Theoretic One-Class Collaborative Filtering Based on Uninteresting Items

Yeon-Chang Lee
Hanyang University, Korea
lyc0324@hanyang.ac.kr

Sang-Wook Kim*
Hanyang University, Korea
wook@hanyang.ac.kr

Dongwon Lee
The Pennsylvania State University, US
dongwon@psu.edu

Abstract

We investigate how to address the shortcomings of the popular *One-Class Collaborative Filtering* (OCCF) methods in handling challenging “sparse” dataset in one-class setting (e.g., clicked or bookmarked), and propose a novel graph-theoretic OCCF approach, named as gOCCF, by exploiting both positive preferences (derived from rated items) as well as negative preferences (derived from unrated items). In capturing both positive and negative preferences as a bipartite graph, further, we apply the *graph shattering* theory to determine the right amount of negative preferences to use. Then, we develop a suite of novel graph-based OCCF methods based on the *random walk with restart* and *belief propagation* methods. Through extensive experiments using 3 real-life datasets, we show that our gOCCF effectively addresses the sparsity challenge and significantly outperforms all of 8 competing methods in accuracy on very sparse datasets while providing comparable accuracy to the best performing OCCF methods on less sparse datasets. The datasets and implementations used in the empirical validation are available for access: <https://goo.gl/sfiawn>.

Introduction

Collaborative filtering (CF) is one of the popular recommendation methods that uses 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) (Adomavicius and Tuzhilin 2005; Cremonesi, Koren, and Turrin 2010; Sarwar et al. 2001). A group of CF methods particularly well suited to handle the one-class setting are known as the *one-class collaborative filtering* (OCCF) method (Pan et al. 2008). In recent years, the interest of the one-class setting has grown substantially and several OCCF methods have been proposed (Pan et al. 2008; Pan and Scholz 2009; He and McAuley 2016; Hu, Koren, and Volinsky 2008; Sindhvani et al. 2010; Rendle et al. 2009; Yao et al. 2014; Ning and Karypis 2011; He et al. 2016; Volkovs and Yu 2015). In general, the one-class setting presents two challenges: (1) one-class setting (e.g., clicked or bookmarked) has inherently *less information* to capture a user’s taste than multi-class setting (e.g., 1-5 star rating); (2) datasets

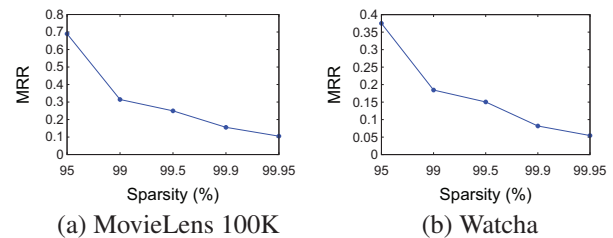


Figure 1: Accuracy of an OCCF method per sparsity.

in one-class setting are often *more sparse* than those in multi-class setting (Pan et al. 2008; Pan and Scholz 2009; Yao et al. 2014).

Despite their success, however, these OCCF methods tend to become less effective in dealing with sparse dataset with many unrated items. To demonstrate this challenge, for instance, consider Figure 1, where the accuracy (in terms of mean reciprocal rank (MRR)) of WRMF (Pan et al. 2008), a popular OCCF method, rapidly drops as the degrees of sparsity of MovieLens 100K and Watcha datasets (We will elaborate this sparsity issue in more detail in a later section. See Table 4 and Figure 8). Therefore, *the goal of this work is to improve existing OCCF methods in handling quite sparse datasets.*

We note, in particular, that the recently proposed *zero-injection*¹ (Hwang et al. 2016) successfully addressed the sparsity problem in *multi-class setting* by finding the so-called *uninteresting items* (*U-items* in short) that users have not rated yet but are unlikely to like even if recommended. The zero-injection identifies $\theta\%$ of *U-items* as *additional negative preferences* to exploit, and injects “0” rating to the identified *U-items* in a user-item matrix. Encouraged by the success of the zero-injection in multi-class setting, first we simply applied it to one-class setting. However, we found that a naive application of the zero-injection poses two new challenges: (1) the recommendation accuracy in one-class setting is *very sensitive* to the parameter θ , which is not the case in multi-class setting; (2) the accuracy of the zero-injection is still inferior to those of existing OCCF methods specifically customized for one-class setting.

¹Note that the zero-injection itself is not an OCCF method but a type of a value imputation method.

*Corresponding author
Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

To address the challenges of the one-class setting and zero-injection, we propose a novel *graph-theoretic OCCF approach*, named as gOCCF, that exploits both positive preferences (*i.e.*, interesting items, I-items) from rated items as well as negative preferences (*i.e.*, U-items) derived from unrated items. Note that this approach *converts the one-class setting to the binary-class setting* with both positive and negative preferences. Furthermore, gOCCF eliminates the use of θ needed in the original zero-injection by determining the effective number of U-items to use via *graph shattering theory* (Appel et al. 2009) and the property of information propagation in a graph.

In addition, to model such binary-class information in a graph, we propose two methods: (1) to model it as *two graphs* independently, taking each of positive and negative preferences separately; (2) to model it as a *single signed graph*, taking both positive and negative preferences together into account. Finally, gOCCF employs the extended versions of *random walk with restart* (RWR) (Shahriari and Jalili 2014; Fouss et al. 2007) and *belief propagation* (BP) (Yedidia, Freeman, and Weiss 2003; Jang et al. 2016) to come up with the final recommendation results.

Our contributions are summarized as follows:

- We demonstrate the limitations of the recently-proposed zero-injection method in handling the one-class setting.
- We design a novel graph-theoretic gOCCF that captures both positive and negative preferences of a user by means of I-items and U-items in graphs. gOCCF is easy to implement (*i.e.*, less than 100 lines code required) on top of existing CF methods.
- We discuss how to determine a right number of U-items (*i.e.*, making gOCCF parameter free) by considering the degree of interestingness for items, graph shattering theory, and the property of information propagation.
- We validate the effectiveness of gOCCF via extensive experiments using three real-life datasets: (1) Utilization of U-items significantly improves the recommendation accuracy of existing graph-based CF methods; (2) gOCCF consistently and universally outperforms all of 8 competing methods in accuracy on very sparse datasets.

Preliminary Results

We first demonstrate the issues that arise when naively applying the state-of-the-art zero-injection (Hwang et al. 2016) to one-class setting rather than multi-class setting. We performed the experiments using CiteULike, a real-world academic literature dataset in one-class setting (See Table 2 in experimental section). As an evaluation, we performed top- N recommendations and used the five measures such as *precision*, *recall*, *normalized discounted cumulative gain* (nDCG) (Järvelin and Kekäläinen 2000), *mean reciprocal rank* (MRR) (Voorhees 1999), and *half-life utility* (HLU) (Breese, Heckerman, and Kadie 1998).

We applied the zero-injection to two popular CF methods (*i.e.*, SVD based CF (Cremonesi, Koren, and Turrin 2010) and PMF based CF (Salakhutdinov and Mnih 2007)). We compared their accuracies with that of a popular OCCF

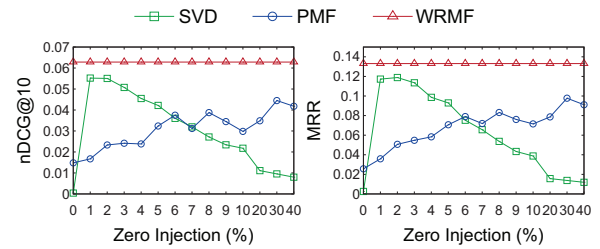


Figure 2: Accuracy in one-class setting: SVD and PMF with varying degree of zero injection, and WRMF without zero injection.

method, WRMF (Pan et al. 2008), to validate whether the zero-injection has a meaningful effect in one-class setting too. Due to space limitations, Figure 2 only shows nDCG@10 and MRR of top- N recommendations by the two CF methods with a different degree (θ)² of the zero-injection and those by WRMF. We increased θ by 1% during the range of 1-10% and by 10% during the range of 10-49%.

Note that (1) similar to multi-class setting, the zero-injection also helps *significantly improve the accuracy* of CF methods in one-class setting; (2) accuracies of CF methods exploiting the zero-injection are *very sensitive* to parameter θ , which determines the number of U-items used in the zero-injection, in one-class setting; (3) accuracies of CF methods exploiting the zero-injection are no better than that of a popular OCCF method, WRMF, targeting at one-class setting. Therefore, blindly applying the state-of-the-art zero-injection to one-class setting will *not necessarily improve* the accuracy over popular OCCF methods.

Proposed Approach

To address the challenges of one-class setting and zero-injection, we propose a novel graph-theoretic OCCF approach, named as gOCCF, that exploits *both* I-items and U-items, modeled as a bipartite graph G of users and items. Further, G with the links between users and I-items are called a *positive* graph, G^+ , and G with the links between users and U-items are called as a *negative* graph, G^- .

Our proposal consists of five steps as follows (Figure 3): (1) we construct a *preference matrix* $P = (p_{i,j})_{m \times n} \in \{0, 1\}^{m \times n}$ of m users and n items such that $p_{i,j} = 1$ if a user i rated an item j , and $p_{i,j} = 0$ otherwise; (2) using P , next, we predict the degree of “interestingness” of items on which users have not rated; (3) we determine the U-items per user by jointly considering the degree of interestingness, graph shattering theory, and information propagation; (4) we model the relationships between the newly derived U-items and users, and those between I-items and users from the social network analysis perspective; (5) finally, we analyze the modeled graph using social network analysis methods and determine top- N recommendations.

²When $\theta=0$, the zero-injection degenerates to the original CF method without utilizing U-items.

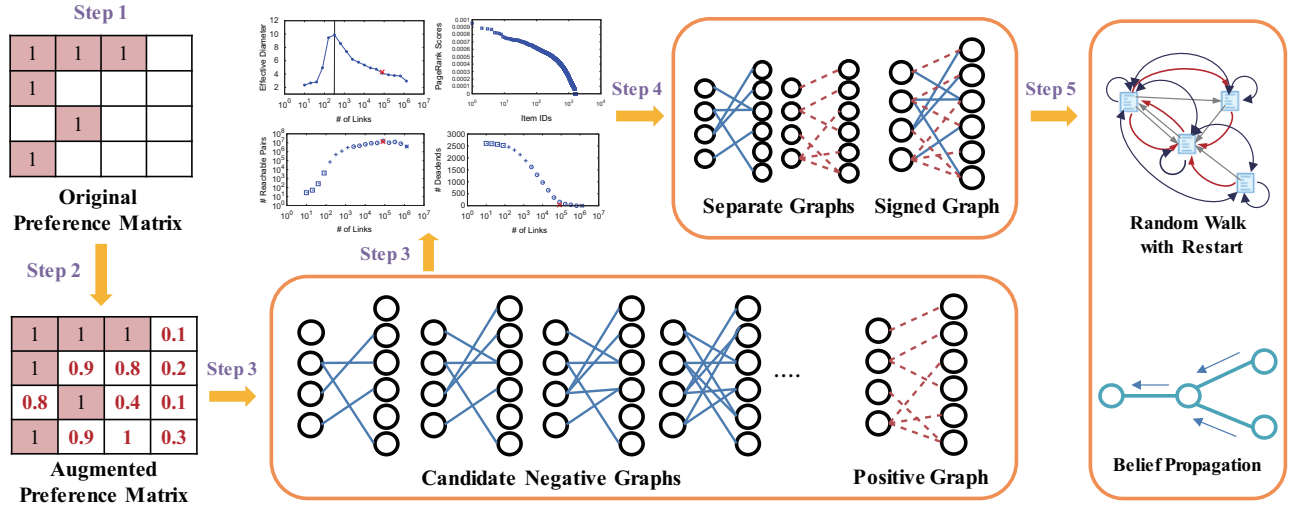


Figure 3: Overview of gOCCF.

Inferring the degree of interestingness

We employ a popular OCCF method, WRMF (Pan et al. 2008), in the same way as the zero-injection to predict the *degree of interestingness on unrated items*³. To do this, we use a preference matrix P and a weight matrix $W = (w_{i,j})_{m \times n} \in [0, 1]^{m \times n}$. We create W by assigning the weights that quantify the relative contribution to the values in P . For rated items with $p_{i,j} = 1$, we assign the highest value of 1 (*i.e.*, $w_{i,j} = 1$) because user i *directly expressed* an opinion for an item j . For unrated items with $p_{i,j} = 0$, we assign the weights between 0 and 1 using the *user-oriented scheme*⁴ proposed in WRMF (Pan et al. 2008). Finally, we approximate P by performing the *weighted alternating least squares* (wALS) method (Srebro and Jaakkola 2003), decomposing P into two low-rank matrices U and V (representing the features of users and items as latent factors, respectively) while optimizing an objective function: $\mathcal{L}(U, V) = \sum_{i,j} w_{i,j} \{(p_{i,j} - U_i V_j^T)^2 + \lambda(\|U_{i(\cdot)}\|_F^2 + \|V_{j(\cdot)}\|_F^2)\}$, where U_i and V_j indicate the i -th row of U , and the j -th row of V , respectively. In addition, $\|\cdot\|_F$ denotes the *Frobenius norm* and λ is a regularization parameter for preventing overfitting.

In order to factorize P , WRMF first assigns random values to elements in V , and updates elements in U as follows: $\forall 1 \leq i \leq m: U_{i(\cdot)} = P_{i(\cdot)} \widetilde{W}_{i(\cdot)} V \{V^T \widetilde{W}_{i(\cdot)} V + \lambda(\sum_j w_{i,j})L\}^{-1}$, where $\widetilde{W}_{i(\cdot)}$ is a diagonal matrix with the elements of $W_{i(\cdot)}$ on the diagonal, and L is an identity matrix. After that, WRMF updates elements in V while fixing U as follows: $\forall 1 \leq j \leq n: V_{j(\cdot)} = P_{(\cdot)j}^T \widetilde{W}_{(\cdot)j} U \{U^T \widetilde{W}_{(\cdot)j} U + \lambda(\sum_i w_{i,j})L\}^{-1}$. We optimize the objective function by repeatedly computing both $U_{i(\cdot)}$ and $V_{j(\cdot)}$ until U and V converge.

³In our comparison of various OCCF methods, we found that WRMF has the best accuracy.

⁴It relies on the intuition that as a user rates more items, she is more likely to dislike unrated items.

Finally, we approximate matrix $\hat{P} = (\hat{p}_{i,j})_{m \times n} \in [0, 1]^{m \times n}$ by calculating an inner product of U and V .

We point out that the accuracy of WRMF deteriorates as dataset becomes sparse. However, note that we *employ WRMF to tackle the problem of choosing U-items rather than that of predicting I-items*. The problem of choosing U-items among relatively many candidates (*i.e.*, unrated items) is *much easier* than that of accurately predicting I-items with only a few correct answers. In our experiment using CiteU-Like, we found that the error rate is only 0.15%, which captures how many *real* I-items are selected as U-items (*i.e.*, mis-classified) by WRMF.

Determining the number of U-items without θ

Next, to avoid the sensitive accuracy to θ that determines *the number of U-items* to use in zero-injection, we determine a right number of U-items by exploiting the degree of interestingness previously computed. First, we can construct a negative graph G^- by gradually increasing the number of selectable U-items, and then find the best performing negative graph G^- by measuring the accuracy of recommendation. In general, however, the search space for the number of U-items is too high. In MovieLens, for instance, there are 1.5M unrated items (*i.e.*, user-item pairs) among which U-items can be chosen. Therefore, it is prohibitively expensive to find the number of U-items with the best accuracy by performing a target recommendation algorithm on *every* negative graph G^- for a given dataset (*i.e.*, dataset and algorithm dependent).

Therefore, we aim to determine the number of U-items *independently* from datasets and recommendation algorithms. We first examine each graph G^- by analyzing graph properties such as topological properties and information propagation. We consider k (unrated) user-item pairs with the lowest degree of interestingness as negative links and model them as a single bipartite graph. In this case, starting from 10, k doubles until negative links reach 90% of unrated user-item pairs.

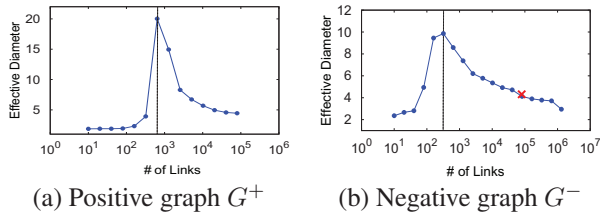


Figure 4: ShatterPlot for G^+ and G^- .

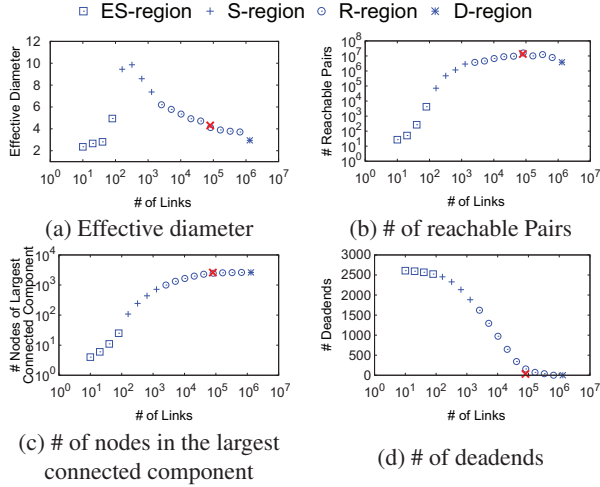


Figure 5: Topological properties for G^+ and G^- .

(1) Topological property. We employ the *graph shattering theory* (Appel et al. 2009) 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. By continuously removing links of a graph, this plot shows the changes in the topological properties of a graph. Figure 4.(a) shows a ShatterPlot for *effective diameters* that change as links are randomly removed from a real positive graph G^+ generated by MovieLens. A vertical line indicates the shattering point.

We examine the change in the topological properties of negative graphs G^- while adding links in the inverse order of degree of interestingness (Figure 4.(b)). Depending on the number of links that are added (*i.e.*, their density), we divide graphs G^- into four regions⁵: extremely sparse region (ES-region), shattered region (S-region), real graph region (R-region), and dense region (D-region). D-region represents a set of graphs much more dense than G^+ , R-region does a set of graphs similar to G^+ in density, S-region represents a set of graphs where shattering starts, and ES-region represents a set of graphs where shattering becomes much more severe,

⁵We will analyze the property of information propagation of negative graphs G^- by using the distribution of PageRank scores for each graph. Note that it is difficult to see those for 18 negative graphs (for MovieLens dataset). We thus analyze the property per region after grouping the graphs with similar density into one region and selecting a representative graph of each region.

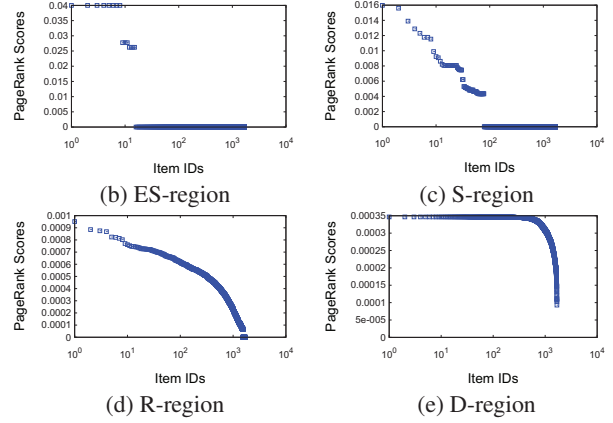
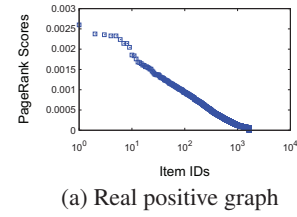


Figure 6: Information propagation for G^+ and the representative graphs in the four regions of G^- .

making a graph extremely sparse.

Figure 5 shows the topological properties of the negative graphs divided into the four regions accordingly, where we note⁶ (1) all ShatterPlots for negative graphs G^- have the shattering points (in particular, the effective diameters have a sharp and clear spike at the shattering point); (2) graphs G^- of R-region with the number of links similar to that of G^+ have *very similar* topological properties to G^+ . In particular, a graph G^- with the number of links *equal to* that of G^+ (shown by ‘x’ around 10^5 in the figure) shows the most similar property.

(2) Information propagation. In this paper, we provide recommendation by performing graph-based methods on a real positive graph G^+ and a negative graph G^- . They analyze the *information propagation* of a given graph and then provide recommendations based on the analysis result. Thus, the effect of *further use of G^-* appears when information propagation of G^+ can be changed by that of G^- .

Therefore, we attempt to find a region of negative graphs G^- that can change the information propagation of G^+ by analyzing each region using PageRank scores known to best characterize the aspect of information propagation. To do this, we first select a representative graph for each region as follows: (1) the graph with the fewest links for ES-region, (2) the graph corresponding to the shattering point for S-region, (3) the graph with the number of links same as that of the real positive graph G^+ for R-region, and (4) the graph with the largest number of links for D-region.

Figure 6 shows PageRank scores⁷ of G^+ and the repre-

⁶The same trend was observed in Watcha and CiteULike.

⁷We set damping factors of PageRank to 0.85 in all cases.

sentative negative graph G^- of each region in MovieLens⁸. From Figure 6, we identify the following observations:

- The distribution of the scores for ES-region and S-region shows that only a small number of nodes (*i.e.*, 0.89% for ES-region and 4.63% for S-region) receive negative scores and the remaining nodes have no negative scores at all. Even if information propagation shown in Figures 6.(b) and (c) is additionally exploited, it is difficult to change the existing propagation because there is a very small number of nodes having negative scores that can change the positive scores of G^+ .
- The distribution of scores for D-region shows that most nodes (*i.e.*, 67.18%) receive very similar negative scores. Even if information propagation shown in Figure 6.(e) additionally exploited, it is difficult to change the existing propagation because most of negative scores that can change the positive scores of G^+ have similar values.
- R-region shows a power law-like distribution with a few hub nodes, similar to a real positive graph G^+ . If information propagation shown in Figure 6.(d) additionally exploited, the existing propagation seems to have significant changes because the negative scores with various values influence the positive scores of G^+ .

These results show that a negative graph G^- with the *same* number of links as a real positive graph G^+ has the topological properties most similar to G^+ , and enables meaningful changes in the information propagation of G^+ when G^+ is used in conjunction with G^- . Therefore, we propose to model G^- by selecting as many U-items as I-items in G^+ . In this section, however, we only show that G^- of R-region can help change the information propagation of G^+ compared to other regions. In other words, we have not confirmed yet whether this change really helps improve recommendation accuracy. In the experimental section, therefore, we confirm that the information propagation changed by our G^- is actually useful to achieve higher accuracy.

Graph-based recommendation

The proposed approach so far enables to transform a dataset in *one-class setting* to one in *binary-class setting* with both U-items and I-items. To model such binary-class information as *undirected graphs*, we consider two methods: (1) two *separate* graphs are modeled by independent consideration of positive and negative links; (2) a *single signed graph* is modeled by taking both positive and negative links together into account. After modeling the graphs, we can perform recommendation by exploiting various graph-based methods. However, since most existing methods are based only on links with positive weights, we extend two popular methods, random walk with restart (RWR) (Fouss et al. 2007) and belief propagation (BP) (Yedidia, Freeman, and Weiss 2003), to consider both positive and negative links (*i.e.*, binary-class setting) with opposite meanings.

(1) RWR based. We perform RWR separately on each graph consisting of only one type of links (Shahriari and Jalili

Table 1: Propagation matrices for BP

(a) $\varphi_{ij}^{(+)}$ for positive links			(b) $\varphi_{ij}^{(-)}$ for negative links		
v_i 's state \ v_j 's state	Interesting	Uninteresting	v_i 's state \ v_j 's state	Interesting	Uninteresting
Interesting	$0.5+\beta$	$0.5-\beta$	Interesting	$0.5-\beta$	$0.5+\beta$
Uninteresting	$0.5-\beta$	$0.5+\beta$	Uninteresting	$0.5+\beta$	$0.5-\beta$

2014), and compute both positive and negative RWR scores for nodes as follows: $\vec{r}^{(+)} = \alpha \bar{w}^{(+)} \vec{r}^{(+)} + (1 - \alpha) \vec{t}$ and $\vec{r}^{(-)} = \alpha \bar{w}^{(-)} \vec{r}^{(-)} + (1 - \alpha) \vec{t}$, where $\vec{r}^{(+)}$ and $\vec{r}^{(-)}$ represent the positive and negative ranking vectors of all nodes, respectively, and $\bar{w}^{(+)}$ and $\bar{w}^{(-)}$ are normalized weight matrices built based on positive and negative links, respectively; α is a damping factor, and \vec{t} is a personalization vector representing the target nodes to be restarted. Next, we compute the final RWR scores of nodes by taking the difference between positive and negative RWR scores:

$$\vec{r} = \vec{r}^{(+)} - \vec{r}^{(-)} \quad (1)$$

We finally recommend top- N items that are most preferred by the target user, based on the calculated ranking vectors.

(2) BP based. Similar to RWR, we first perform BP separately on each graph consisting of only one type of links. BP infers the state of a node in a graph by computing the belief score of the node through exchanging messages between nodes. The message is a node's opinion about its adjacent node's possibility of being in a specific state. In our case, the states of a node are binary – *i.e.*, *interesting* or *uninteresting*. For positive and negative graphs, the message sent from one node to its neighbor is represented as a vector and computed as follows: $m_{ij}^{(+)}(\sigma) \leftarrow \sum \phi_i^{(+)}(\sigma') \varphi_{ij}^{(+)}(\sigma', \sigma) \prod_{k \in N(i) \setminus j} m_{ki}^{(+)}(\sigma)$ and $m_{ij}^{(-)}(\sigma) \leftarrow \sum \phi_i^{(-)}(\sigma') \varphi_{ij}^{(-)}(\sigma', \sigma) \prod_{k \in N(i) \setminus j} m_{ki}^{(-)}(\sigma)$, where $m_{ij}^{(+)}(\sigma)$ (resp. $m_{ij}^{(-)}(\sigma)$) denotes the message from v_i to v_j , saying v_i 's belief about v_j 's likelihood of being in state σ for G^+ (resp. G^-). The message from v_i to v_j is made up with the product of the message from v_i 's adjacent nodes except v_j ($N(i) \setminus j$), where $N(i)$ represents a set of v_i 's adjacent nodes. $\phi_i^{(+)}(\sigma')$ (resp. $\phi_i^{(-)}(\sigma')$) is an observed prior state that represents the probability of v_i being in state σ' for G^+ (resp. G^-), and $\varphi_{ij}^{(+)}(\sigma', \sigma)$ (resp. $\varphi_{ij}^{(-)}(\sigma', \sigma)$) represents the probability of v_j being in state σ when its adjacent node v_j is in state σ' for G^+ (resp. G^-). The propagation matrices $\varphi_{ij}^{(+)}$ and $\varphi_{ij}^{(-)}$ can be given as in Table 1 where $0 < \beta \ll 1$.

Message passing repeats until a change in message scores becomes below a given threshold. Afterward, for positive and negative graphs, the belief score is computed as follows, respectively: $b_i^{(+)}(\sigma) = k \prod_{j \in N(i)^{(+)}} m_{ji}^{(+)}(\sigma)$ and $b_i^{(-)}(\sigma) = k \prod_{j \in N(i)^{(-)}} m_{ji}^{(-)}(\sigma)$, where $b_i^{(+)}(\sigma)$ (resp. $b_i^{(-)}(\sigma)$) represents the probability that a node i will be interesting (resp. uninteresting) to a target user using G^+ (resp. G^-); $N(i)^{(+)}$ (resp. $N(i)^{(-)}$) refers to a set of adja-

⁸The same trend was observed in Watcha and CiteULike.

Table 2: Dataset statistics

Datasets	# users	# items	# user-item pairs	Sparsity
MovieLens	943	1,682	100,000	93.69%
Watcha	1,391	1,927	100,000	96.98%
CiteULike	5,551	16,980	210,504	99.82%

cent nodes to node i in G^+ (resp. G^-). We compute the final belief score of each node by using the difference between positive and negative belief scores for the node as follows:

$$b_i(\sigma) = b_i^{(+)}(\sigma) - b_i^{(-)}(\sigma) \quad (2)$$

Finally, we recommend top- N items that a target user likes the most, based on the final belief scores.

We further suggest a way to perform BP on a *signed graph* that contains both positive and negative links together. In this case, message passing between nodes according to positive and negative links is performed as follows: $m_{ij}(\sigma) \leftarrow (1) \sum \phi_i(\sigma') \varphi_{ij}^{(+)}(\sigma', \sigma) \prod_{k \in N(i) \setminus j} m_{ki}(\sigma')$ if i and j are connected by positive links; or (2) $\sum \phi_i(\sigma') \varphi_{ij}^{(-)}(\sigma', \sigma) \prod_{k \in N(i) \setminus j} m_{ki}(\sigma')$ if i and j are connected by negative links, where $\varphi_{ij}^{(+)}$ (resp. $\varphi_{ij}^{(-)}$) represents the propagation matrix applied to G^+ (resp. G^-); $N(i)$ denotes a set of adjacent nodes to node i in the signed graph. The computation of the belief score for a node v_i 's probability of being in state σ is as follows:

$$b_i(\sigma) = k \prod_{j \in N(i)} m_{ji}(\sigma) \quad (3)$$

Finally, we recommend top- N items that a target user likes the most, based on the final belief scores.

Empirical Validation

Experimental Settings

(1) **Datasets.** We use 3 real-life datasets: MovieLens, Watcha, and CiteULike. As both MovieLens and Watcha contain movie ratings of 1-5 range (*i.e.*, multi-class setting), we convert their 1-5 ratings into 1 (*i.e.*, one-class setting), as popularly done in other OCCF researches (Rendle et al. 2009; He et al. 2016; Volkovs and Yu 2015; Ning and Karypis 2011). CiteULike contains only implicit feedback, thus is originally in one-class setting. Table 2 shows the details.

(2) **Evaluation.** We perform top- N recommendations and employ five measures of Precision, Recall, nDCG (Järvelin and Kekäläinen 2000), MRR (Voorhees 1999), and HLU (Breese, Heckerman, and Kadie 1998) to evaluate the accuracy of recommendations using 5-cross validation.

(3) **Competing methods.** We compare gOCCF with 1 baseline and 8 state-of-the-art methods to verify its effectiveness. As the baseline, we first compare ours with a non-personalized method using item popularity, called MostPopular. Second, we compare ours with two graph-based methods: RWR (Fouss et al. 2007) and BP (Yedidia,

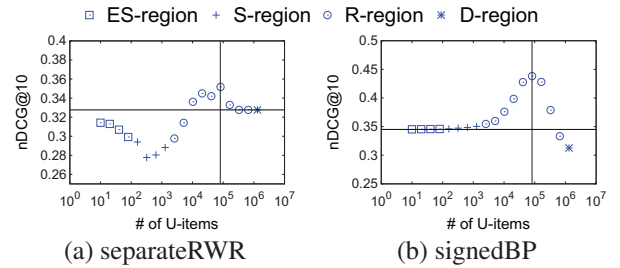


Figure 7: Accuracy of separateRWR and signedBP according to the number of U-items.

Freeman, and Weiss 2003). Third, we compare ours with two zero-injection (Hwang et al. 2016) added CF methods: SVD_ZI (Cremonesi, Koren, and Turrin 2010), and PMF_ZI (Salakhutdinov and Mnih 2007). Lastly, we compare ours with four OCCF methods: WRMF (Pan et al. 2008), BPRMF (Rendle et al. 2009), GBPRMF (Pan and Chen 2013), and SLIM (Ning and Karypis 2011). We use the implementations of those methods – *i.e.*, SVD_ZI, PMF_ZI, WRMF, BPRMF, GBPRMF, and SLIM – in the open-source MyMediaLite (Gantner et al. 2011), Graphchi (Kyrola, Blelloch, and Guestrin 2012), and Librec (Guo et al. 2015) and employ their best performing parameter settings in each dataset. gOCCF includes the following three variants: (1) separateRWR: our RWR method using two separate graphs using Eq. (1); (2) separateBP: our BP method using two separate graphs using Eq. (2); (3) signedBP: our BP method using a signed graph using Eq. (3).

(4) **Parameter settings.** The optimal parameter settings of gOCCF methods were determined by our extensive experiments. The details are as follows: (1) separateRWR: the number of iterations and the damping factor are 5 and 0.85, respectively. (2) separateBP: $\phi_i^{(+)}(\sigma = interesting)^9$ is 0.9 if u has evaluated i and 0.5 otherwise. $\phi_i^{(-)}(\sigma = interesting)$ is 0.4 if i is U-items of u , and 0.5 otherwise. (3) signedBP: $\phi_i(\sigma = interesting)$ is 0.9 if u has evaluated i , 0.4 if i is U-items of u , and 0.5 otherwise. In separateRWR and separateBP, the number of iterations and β of propagation matrices (Table 1) are set as 5 and 0.0001, respectively.

Experimental Results

(1) **Effectiveness of our U-item decision.** We first conduct an experiment to verify whether a negative graph G^- modeled by our decision method helps improve recommendation accuracy. Figure 7 shows top- N recommendation accuracies of separateRWR and signedBP according to the number of U-items in MovieLens (Due to space limitations, we omit the results for separateBP). The vertical line shows the number of links in a real positive graph G^+ and the horizontal line shows the accuracy of the original RWR and BP without using the idea of U-items. The results show that all of our methods with two separate graphs and a signed graph always have the best accuracy when they use negative graphs

⁹We present only the value for *interesting* because that for the *uninteresting* can simply be computed as $1 - \text{value}(\text{interesting})$

Table 3: Accuracy of graph-based methods before and after exploiting U-items

Metrics	MovieLens		
	separateRWR	separateBP	signedBP
P@10	0.302 (9.2%)	0.309 (7.2%)	0.370 (28.2%)
R@10	0.171 (8.2%)	0.164 (10.9%)	0.210 (42.1%)
nDCG@10	0.352 (7.3%)	0.365 (5.7%)	0.438 (27.0%)
MRR	0.584 (2.5%)	0.604 (1.9%)	0.679 (14.6%)
HLU	43.894 (2.8%)	47.195 (1.4%)	55.980 (20.3%)
Metrics	Watcha		
	separateRWR	separateBP	signedBP
P@10	0.113 (10.8%)	0.124 (13.2%)	0.151 (38.3%)
R@10	0.107 (9.6%)	0.113 (13.4%)	0.142 (41.9%)
nDCG@10	0.136 (9.9%)	0.152 (13.2%)	0.190 (41.8%)
MRR	0.295 (7.1%)	0.329 (10.8%)	0.391 (32.0%)
HLU	14.288 (7.7%)	17.448 (20.4%)	23.012 (58.8%)
Metrics	CiteULike		
	separateRWR	separateBP	signedBP
P@10	0.122 (21.0%)	0.112 (26.9%)	0.091 (2.9%)
R@10	0.199 (13.5%)	0.162 (26.4%)	0.131 (2.4%)
nDCG@10	0.202 (18.2%)	0.175 (29.9%)	0.138 (2.3%)
MRR	0.326 (14.0%)	0.295 (21.0%)	0.247 (1.3%)
HLU	21.257 (23.0%)	19.323 (33.3%)	14.680 (1.2%)

G^- belonging to R-region. In particular, they provide the highest accuracy when having the same number of negative links as that of positive links. As such, in subsequent experiments, we set the number of negative links as the same as the number of positive links.

In Figure 7, other regions excluding R-region show different trends per graph modeling. For separate graphs (Figure 7.(a)), ES-region and S-region exhibit reduction in accuracy, while D-region shows little difference from original RWR in accuracy. In particular, S-region seems to have wrong information propagation as a graph shatters and the connectivity of the graph collapses. In D-region where most of items have similar negative RWR scores in separateRWR, the scores have little effect on the final scores. For the signed graph (Figure 7.(b)), unlike separate graphs, both ES-region and S-region show little difference from original BP in accuracy, but D-region shows a large drop in accuracy. This is because only a small number of negative links are added in both ES-region and S-region. In D-region, as many negative links are added, the scores of most items are dominated by such negative links.

(2) Accuracy of graph-based methods exploiting U-items. We conduct an experiment to confirm whether the utilization of U-items in existing graph-based methods helps improve recommendation accuracy. To do this, we compare the accuracies of graph-based methods *before* and *after* exploiting U-items. We employ two testing item sets for comparison: all items and long-tail items – *i.e.*, items with only a few ratings excluding p popular items ($p = 100$ for MovieLens and Watcha, and $p = 1, 452$ for CiteULike) that have 30% of total ratings in each dataset. Table 3 shows the results in all items setting (Due to space limitations, we omit the

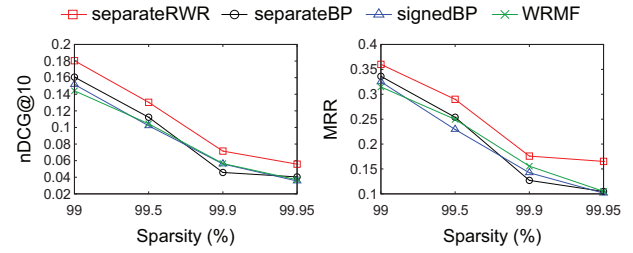


Figure 8: Accuracy of gOCCF methods and WRMF per sparsity.

results for long-tail items, which are consistent with these in Table 3.). We confirm that utilizing U-items in gOCCF is effective in improving the accuracy significantly. This result is consistently shown in all measures and all datasets used. Furthermore, for both separateRWR and separateBP, Table 3 shows that accuracy improves as datasets become more sparse (See the data sparsity shown in Table 2). For signedBP, both MovieLens and Watcha show a significant improvement while CiteULike shows a relatively small improvement. Through these experiments, we conclude that our gOCCF family utilizing U-items significantly improves the accuracy of existing graph-based methods and is also robust even for highly sparse datasets.

(3) Comparisons with the state-of-the-art methods. We perform an experiment to compare the recommendation accuracies among: (1) a baseline method, (2) three zero-injection methods, (3) four OCCF methods, and (4) three gOCCF methods. Table 4 shows the results using three datasets¹⁰. First, in the most dense MovieLens, signedBP outperforms all other methods except WRMF and GBPRMF. In the more sparse Watcha, however, signedBP outperforms all other methods. Finally, *in CiteULike with the real one-class setting, separateRWR and separateBP of gOCCF outperform all the other methods.* It also shows that, as a dataset becomes more sparse, the accuracy reduction of competing methods grows much larger than that of our gOCCF.

Since MovieLens is relatively dense compared to the typical datasets in one-class setting, we performed the same experiment by adjusting the sparsity of MovieLens. Figure 8 shows the results of measuring the accuracy of WRMF and gOCCF, adjusting the sparsity of MovieLens to 99.95%, 99.90%, 99.50%, and 99.00%. Note that despite the increase of sparsity, gOCCF maintains its lead over WRMF. In other words, gOCCF is robust to such a sparse dataset because it gathers sufficient information via both U-items and I-items, and further helps densify a user-item rating matrix. This tendency repeats in all other measures and in the experiment with Watcha.

Related Work

Most OCCF methods view *all* unrated items as negative preferences (*i.e.*, all missing as negative (AMAN) concept)

¹⁰In CiteULike, as SLIM crashed with out-of-memory, we omit it from the experiment.

Table 4: Accuracy of 7 competing methods and gOCCF methods

Metrics	MovieLens									
	MostPopular	SVD_ZI	PMF_ZI	WRMF	BPRMF	GBPRMF	SLIM	separateRWR	separateBP	signedBP
P@10	0.222	0.365	0.343	0.393	0.368	0.385	0.337	0.302	0.309	0.370
R@10	0.113	0.203	0.198	0.222	0.215	0.220	0.194	0.171	0.164	0.210
nDCG@10	0.250	0.424	0.389	0.460	0.435	0.454	0.396	0.352	0.365	0.438
MRR	0.453	0.652	0.589	0.689	0.679	0.692	0.630	0.584	0.604	0.679
HLU	30.600	51.971	42.921	56.132	54.731	56.747	49.097	43.894	47.195	55.980
Watcha										
	MostPopular	SVD_ZI	PMF_ZI	WRMF	BPRMF	GBPRMF	SLIM	separateRWR	separateBP	signedBP
P@10	0.090	0.141	0.137	0.146	0.120	0.121	0.123	0.113	0.124	0.151
R@10	0.085	0.134	0.136	0.140	0.110	0.115	0.117	0.107	0.113	0.142
nDCG@10	0.108	0.170	0.165	0.182	0.181	0.150	0.152	0.136	0.152	0.190
MRR	0.253	0.346	0.335	0.375	0.376	0.305	0.328	0.295	0.329	0.391
HLU	11.860	18.181	16.884	21.077	21.929	17.299	17.453	14.288	17.448	23.012
CiteULike										
	MostPopular	SVD_ZI	PMF_ZI	WRMF	BPRMF	GBPRMF	SLIM	separateRWR	separateBP	signedBP
P@10	0.012	0.043	0.034	0.045	0.092	0.049	-	0.122	0.112	0.091
R@10	0.029	0.044	0.037	0.049	0.140	0.078	-	0.199	0.162	0.131
nDCG@10	0.023	0.055	0.044	0.062	0.136	0.066	-	0.202	0.175	0.138
MRR	0.050	0.117	0.073	0.133	0.240	0.132	-	0.326	0.295	0.247
HLU	1.527	5.899	4.489	7.198	12.754	4.005	-	21.257	19.323	14.680

in two camps (He et al. 2016): (1) whole-data based learning (e.g., WRMF) (Pan et al. 2008; Pan and Scholz 2009; Hu, Koren, and Volinsky 2008; Sindhvani et al. 2010; Yao et al. 2014; He et al. 2016; Volkovs and Yu 2015) and (2) sample based learning (e.g., BPRMF) (Rendle et al. 2009; Pan and Chen 2013; Rendle and Freudenthaler 2014). The whole-data based learning determines the probability that a user’s unrated item is a negative preference. Next, the sample-based learning samples negative items randomly or by a specific criterion (e.g., popularity). However, they only exploit the rated items to sample negative items among unrated items or determine probabilities of unrated items being negative items. Therefore, if a dataset is sparse (i.e., only a few rated items in a user-item matrix), it becomes difficult to sample the proper negative items or determine the right probabilities for being negative items.

To alleviate this data sparsity problem, on the other hand, several data imputation methods (Ma, King, and Lyu 2007; Hwang et al. 2016) have been proposed in multi-class setting. They first predict the ratings for the items that were not evaluated, and densify the dataset by filling the predicted ratings in a user-item matrix. In particular, zero-injection (Hwang et al. 2016) finds an individual user’s U-items among unrated items and imputes zero values as their ratings. However, as demonstrated earlier, for one-class datasets, zero-injection suffers from lower accuracy than existing OCCF methods and the sensitivity for the number of U-items.

Conclusions

To address the sparsity issue in one-class setting, we proposed a novel graph-theoretic gOCCF that utilizes the con-

cept of uninteresting items (i.e., U-items). To further alleviate the zero-injection’s shortcomings in one-class setting, gOCCF includes a parameter-agnostic method to determine a right number of U-items by considering the degree of interestingness for unrated items, graph shattering theory, and property of information propagation. We also presented three extended variations, separateRWR, separateBP, and signedBP, based on RWR and BP, to provide accurate recommendation. From comprehensive experiments, we demonstrated that gOCCF outperforms all existing OCCF methods in accuracy on sparse datasets. Applied to CiteULike, for instance, gOCCF improves nDCG@10, MRR, and HLU by up to 48%, 36%, and 67% over the best-performing BPRMF, among existing OCCF methods.

Acknowledgment

This work was supported by the NRF grant funded by the MSIT of Korea (NRF-2017R1A2B3004581), the Next-Generation Information Computing Development Program through the NRF funded by the MSIT of Korea (NRF-2017M3C4A7083678), and the NSF CNS-1742702 award. This work was also supported financially by the Naver Corporation.

References

- Adomavicius, G., and Tuzhilin, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng. (TKDE)* 17(6):734–749.
- Appel, A. P.; Kumar, R.; Chakrabarti, D.; Leskovec, J.; Faloutsos, C.; and Tomkins, A. 2009. Shatterplots: Fast

- tools for mining large graphs. In *Proc. of the SIAM Int'l Conf. on Data Mining (SDM)*, 802–813.
- Breese, J. S.; Heckerman, D.; and Kadie, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. of the Int'l Conf. on Uncertainty in Artificial Intelligence (UAI)*, 43–52.
- Cremonesi, P.; Koren, Y.; and Turrin, R. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proc. of the ACM Int'l Conf. on Recommender Systems (RecSys)*, 39–46.
- Fouss, F.; Pirotte, A.; Renders, J. M.; and Saerens, M. 2007. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Trans. Knowl. Data Eng. (TKDE)* 19(3):355–369.
- Gantner, Z.; Rendle, S.; Freudenthaler, C.; and Schmidt-Thieme, L. 2011. Mymedialite: A free recommender system library. In *Proc. of the ACM Int'l Conf. on Recommender Systems (RecSys)*, 305–308.
- Guo, G.; Zhang, J.; Sun, Z.; and Yorke-Smith, N. 2015. Librecc: A java library for recommender systems. In *Workshop Proc. of the Int'l Conf. on User Modeling, Adaptation, and Personalization (UMAP Workshop)*, volume 1388.
- He, R., and McAuley, J. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proc. of the Int'l Conf. on World Wide Web (WWW)*, 507–517.
- He, X.; Zhang, H.; Kan, M.-Y.; and Chua, T.-S. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proc. of the ACM Int'l Conf. on Research and Development in Information Retrieval (SIGIR)*, 549–558.
- Hu, Y.; Koren, Y.; and Volinsky, C. 2008. Collaborative filtering for implicit feedback datasets. In *Proc. of the IEEE Int'l Conf. on Data Mining (ICDM)*, 263–272.
- Hwang, W. S.; Parc, J.; Kim, S. W.; Lee, J.; and Lee, D. 2016. “told you i didn’t like it”: Exploiting uninteresting items for effective collaborative filtering. In *Proc. of the IEEE Int'l Conf. on Data Engineering (ICDE)*, 349–360.
- Jang, M. H.; Faloutsos, C.; Kim, S. W.; Kang, U.; and Ha, J. 2016. Pin-trust: Fast trust propagation exploiting positive, implicit, and negative information. In *Proc. of the ACM Int'l Conf. on Information and Knowledge Management (CIKM)*, 629–638.
- Järvelin, K., and Kekäläinen, J. 2000. Ir evaluation methods for retrieving highly relevant documents. In *Proc. of the ACM Int'l Conf. on Research and Development in Information Retrieval (SIGIR)*, 41–48.
- Kyrola, A.; Blelloch, G.; and Guestrin, C. 2012. Graphchi: Large-scale graph computation on just a pc. In *Proc. of the USENIX Symp. on Operating Systems Design and Implementation (OSDI)*, volume 12, 31–46.
- Ma, H.; King, I.; and Lyu, M. R. 2007. Effective missing data prediction for collaborative filtering. In *Proc. of the ACM Int'l Conf. on Research and Development in Information Retrieval (SIGIR)*, 39–46.
- Ning, X., and Karypis, G. 2011. Slim: Sparse linear methods for top-n recommender systems. In *Proc. of the IEEE Int'l Conf. on Data Mining (ICDM)*, 497–506.
- Pan, W., and Chen, L. 2013. Gbpr: Group preference based bayesian personalized ranking for one-class collaborative filtering. In *Proc. of the Int'l Joint Conf. on Artificial Intelligence (IJCAI)*, volume 13, 2691–2697.
- Pan, R., and Scholz, M. 2009. Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering. In *Proc. of the ACM Int'l Conf. on Knowledge Discovery and Data Mining (KDD)*, 667–676.
- Pan, R.; Zhou, Y.; Cao, B.; Liu, N. N.; Lukose, R.; Scholz, M.; and Yang, Q. 2008. One-class collaborative filtering. In *Proc. of the IEEE Int'l Conf. on Data Mining (ICDM)*, 502–511.
- Rendle, S., and Freudenthaler, C. 2014. Improving pairwise learning for item recommendation from implicit feedback. In *Proc. of the Int'l Conf. on Web Search and Data Mining (WSDM)*, 273–282.
- Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. Bpr: Bayesian personalized ranking from implicit feedback. In *Proc. of the Int'l Conf. on Uncertainty in Artificial Intelligence (UAI)*, 452–461.
- Salakhutdinov, R., and Mnih, A. 2007. Probabilistic matrix factorization. In *Proc. of the Intl Conf. on Neural Information Processing Systems (NIPS)*, volume 1, 2–1.
- Sarwar, B.; Karypis, G.; Konstan, J.; and Riedl, J. 2001. Item-based collaborative filtering recommendation algorithms. In *Proc. of the Int'l Conf. on World Wide Web (WWW)*, 285–295.
- Shahriari, M., and Jalili, M. 2014. Ranking nodes in signed social networks. *Social Network Analysis and Mining* 4(1):172.
- Sindhwani, V.; Bucak, S. S.; Hu, J.; and Mojsilovic, A. 2010. One-class matrix completion with low-density factorizations. In *Proc. of the IEEE Int'l Conf. on Data Mining (ICDM)*, 1055–1060.
- Srebro, N., and Jaakkola, T. 2003. Weighted low-rank approximations. In *Proc. of the Intl Conf. on Machine Learning (ICML)*, volume 3, 720–727.
- Volkovs, M., and Yu, G. W. 2015. Effective latent models for binary feedback in recommender systems. In *Proc. of the ACM Int'l Conf. on Research and Development in Information Retrieval (SIGIR)*, 313–322.
- Voorhees, E. M. 1999. The trec-8 question answering track report. In *Proc. of the Text REtrieval Conference (TREC-8)*, volume 99, 77–82.
- Yao, Y.; Tong, H.; Yan, G.; Xu, F.; Zhang, X.; Szymanski, B. K.; and Lu, J. 2014. Dual-regularized one-class collaborative filtering. In *Proc. of the ACM Int'l Conf. on Information and Knowledge Management (CIKM)*, 759–768.
- Yedidia, J.; Freeman, W.; and Weiss, Y. 2003. Understanding belief propagation and its generalizations. *Exploring Artificial Intelligence in the New Millennium* 8:236–239.