Personalized Graph Signal Processing for Collaborative Filtering

Jiahao Liu*
School of Computer Science
Fudan University
Shanghai, China
jiahaoliu21@m.fudan.edu.cn

Dongsheng Li
Microsoft Research Asia
Shanghai, China
dongshengli@fudan.edu.cn

Hansu Gu†
Seattle, United States
hansug@acm.org

Tun Lu*†
School of Computer Science
Fudan University
Shanghai, China
lutun@fudan.edu.cn

Peng Zhang*
School of Computer Science
Fudan University
Shanghai, China
zhangpeng_@fudan.edu.cn

Li Shang*
School of Computer Science
Fudan University
Shanghai, China
lishang@fudan.edu.cn

Ning Gu*
School of Computer Science
Fudan University
Shanghai, China
ninggu@fudan.edu.cn

ABSTRACT

The collaborative filtering (CF) problem with only user-item interaction information can be solved by graph signal processing (GSP), which uses low-pass filters to smooth the observed interaction signals on the similarity graph to obtain the prediction signals. However, the interaction signal may not be sufficient to accurately characterize user interests and the low-pass filters may ignore the useful information contained in the high-frequency component of the observed signals, resulting in suboptimal accuracy. To this end, we propose a personalized graph signal processing (PGSP) method for collaborative filtering. Firstly, we design the personalized graph signal containing richer user information and construct an augmented similarity graph containing more graph topology information, to more effectively characterize user interests. Secondly, we devise a mixed-frequency graph filter to introduce useful information in the high-frequency components of the observed signals by combining an ideal low-pass filter that smooths signals globally and a linear low-pass filter that smooths signals locally. Finally, we combine the personalized graph signal, the augmented similarity graph and the mixed-frequency graph filter by proposing a pipeline consisting of three key steps: pre-processing, graph convolution and post-processing. Extensive experiments show that PGSP can achieve superior accuracy compared with state-of-the-art CF methods and, as a nonparametric method, PGSP has very high training efficiency.

CCS CONCEPTS

• **Information systems** → **Personalization**.

KEYWORDS

recommendation, collaborative filtering, graph signal processing

ACM Reference Format:
Jiahao Liu, Dongsheng Li, Hansu Gu, Tun Lu, Peng Zhang, Li Shang, and Ning Gu. 2023. Personalized Graph Signal Processing for Collaborative Filtering. In *Proceedings of the ACM Web Conference 2023 (WWW '23), April 30–May 04, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3543507.3583466

1 INTRODUCTION

Collaborative filtering (CF) is one of the most popular techniques in recommender systems, which predicts the interaction possibility between users and items through interaction history [11–13, 15, 25, 26]. Recently, graph signal processing (GSP) has been applied in CF tasks [4, 18, 21, 22, 27, 31], which can transform the interaction signals from spatial domain to spectral domain through Graph Fourier Transform (GFT) to capture the structure information of the whole similarity graph. Generally, GSP-based CF methods first construct a similarity graph between items, then use a low-pass filter defined on the similarity graph to smooth the users' interaction signals to obtain the prediction signals. Several recent studies [14, 23, 27] show that these methods are related to both neighborhood-based CF methods [13, 26] and GCN-based CF methods [11].

Existing GSP-based CF methods rely on the following two assumptions: 1) a user's personalized preference is completely characterized by his/her interaction history and 2) the low-frequency information in the interaction signal is enough to predict user

*Also with Shanghai Key Laboratory of Data Science, Fudan University, Shanghai, China.
†Corresponding author.

WWW '23, April 30–May 04, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9416-1/23/04…$15.00
https://doi.org/10.1145/3543507.3583466

preferences. However, due to data sparsity issues in many recommender systems, sparse historical interactions may not be enough to accurately describe user preferences. Besides, only using low-frequency information containing common user interests may not be personalized enough due to neglecting the useful high-frequency information containing unique user interests.

In this paper, we propose the *personalized graph signal*, a signal with richer user information, to describe users more accurately. Instead of only using the interaction information of users, the personalized graph signals use both the similarity information between users-items and the similarity information between users-users to describe users. Moreover, we construct an *augmented similarity graph* to utilize the signals more effectively. The augmented similarity graph contains not only the similarity information between items-items, but also the similarity information between users-items and the similarity information between users-users. With the personalized graph signal and the augmented similarity graph, users who are similar to the target user can amplify the signals of the items they have interacted with, which often contain the potential interests of the target user. We construct the similarity relationship from the perspective of random walk, which shares some of the motivations in Node2Vec [8].

In the real world, the observed interaction signal of each user is a mixture of the real preference signal dominated by low-frequency information and the noise signal dominated by high-frequency information. Specifically, the real preferences of each user are composed of globally smooth signals that reflect the general user preferences and locally smooth but globally rough signal that reflects the personalized user preferences. In order to obtain these two kinds of signals, we propose a *mixed-frequency graph filter*, which is a combination of an ideal low-pass filter that smooths signals globally and a linear low-pass filter that smooths signals locally. By controlling the proportion of the two signals, we can obtain more accurate recommendation results than using only an ideal low-pass graph filter.

Finally, we propose the *personalized graph signal processing* (PGSP) method by designing a pipeline to combine the personalized graph signal, the augmented similarity graph and the mixed-frequency graph filter to achieve higher accuracy. PGSP consists of three key steps: *1) pre-processing*, in which we construct the personalized graph signal by concatenating user similarity information with user-item interaction signal; *2) graph convolution*, in which we use the proposed mixed frequency graph filter to obtain both globally smooth signals and locally smooth but globally rough signals and *3) post-processing*, in which we recover the predicted interaction signal. Extensive experiments show that PGSP has superior performance in prediction accuracy. Meanwhile, we analyze the role of the mixed-frequency graph filter and show that, as a nonparametric method, PGSP has very high training efficiency.

The main contributions of this work are summarized as follows:

- We propose the personalized graph signal and the augmented similarity graph for GSP-based CF, which can characterize user interests more effectively and realize more personalized recommendations.
- We reveal the effectiveness of high-frequency components in the observed signal, and find that controlling the ratio of globally smooth signal to locally smooth signal by the proposed mixed-frequency graph filter can further improve the accuracy.
- We propose the personalized graph signal processing (PGSP) method by combining the personalized graph signal, the augmented similarity graph and mixed-frequency graph filters. Experiments show that PGSP can achieve higher accuracy compared with state-of-the-art CF methods.

2 RELATED WORK

GSP aims to develop tools for processing data defined on irregular graph domains, including sampling, filtering and graph learning [5, 24]. Next, we briefly summarize GSP-based CF works.

Most GSP-based methods treat the CF problem as graph signal reconstruction [1]. Narang et al. [21] obtain the interpolated signal by projecting the input signal into the appropriate bandlimited graph signal space. Then, Narang et al. [22] extend the previous work to an iterative method, which has higher computational efficiency and takes into account the reconstruction error relative to the known signals. Wang et al. [31] reconstruct the graph signal by reweighting the sampled residuals for different vertices or propagating the sampled residuals in their respective local sets. Chen et al. [4] formulate graph signal recovery as an optimization problem and provide a general solution through the alternating direction methods of multipliers. Ma et al. [18] interpret the smoothing of the signal by low-pass filtering operation as the diffusion of temperature and prove that the diffused signals are stable to perturbations in the underlying network.

Some CF methods introduced GSP into neural networks. SpectralCF [37] constructs a deep model for learning in the spectral domain, which uses rich connection information to alleviate the cold start problem. Zhang et al. [36] extend SpectralCF by directly mining the association rules between the target user and the target item through FP-Growth [9], and then solve the problems of cold start and data sparsity. AGE [6] proposes an attributed graph embedding framework, which consists of a carefully-designed Laplacian smoothing filter and an adaptive encoder. Yu and Qin [35] propose a low-pass graph filter to remove the noise and reduce the complexity of graph convolution in an unscathed way. AGCN [7] further reduces complexity with Chebyshev polynomial graph filters.

Recently, some works have shown that there are close relationships between GSP-based CF methods and other CF methods [17, 34]. Huang et al. [14] demonstrate that neighborhood-based methods can be modeled as a specific band-stop graph filter and low-rank matrix completion can be viewed as band-limited interpolation algorithms. Nt and Maehara [23] find that graph neural networks only perform low-pass filtering on feature vectors and do not have the non-linear manifold learning property, which means that the graph structure only provides a means to denoise the data. Wu et al. [32] reduce the complexity of GCN through successively removing nonlinearities and collapsing weight matrices between consecutive layers, and show that the resulting linear model corresponds to a fixed low-pass filter. GF-CF [27] develops a unified graph convolution-based framework for CF and proves that many existing CF methods are special cases of the framework which correspond to different low-pass filters in GSP.

3 PRELIMINARIES

This section introduces the background knowledge of GSP and the motivation for using GSP in CF tasks.

3.1 Graph Signal Processing

Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_1, v_2, ..., v_n\}$ is the vertex set with n nodes, $\mathcal{E}$ is the edge set. The topology structure of graph $\mathcal{G}$ can be denoted by a weighted adjacency matrix $W = \{w_{ij}\} \in \mathbb{R}^{n\times n}$, where $w_{ij} > 0$ if $(v_i, v_j) \in \mathcal{E}$, indicating the similarity between v_i and v_j, otherwise $w_{ij} = 0$. $D = diag(d_1, d_2, ..., d_n) \in \mathbb{R}^{n\times n}$ denotes the degree matrix of W, where $d_i = \sum_{v_j\in\mathcal{V}} w_{ij}$ is the degree of node v_i. The graph Laplacian matrix is defined as $L = D - W$.

The signal on the node v_i is defined as a mapping $x_i : \mathcal{V} \to \mathbb{R}$. We can take $\boldsymbol{x} \in \mathbb{R}^n$ as a graph signal where each node is assigned with a scalar. The energy of the graph signal is defined as $E(\boldsymbol{x}) = ||\boldsymbol{x}||^2$. The smoothness of the graph signal can be measured by the total variation as follows:

$$TV(\boldsymbol{x}) = \boldsymbol{x}^T L\boldsymbol{x} = \sum_{(v_i,v_j)\in\mathcal{E}} w_{ij}(x_i - x_j)^2.$$

The normalized total variation of $\boldsymbol{x}$ can be calculated with the Rayleigh quotient as follows:

$$Ray(\boldsymbol{x}) = \frac{TV(\boldsymbol{x})}{E(\boldsymbol{x})} = \frac{\boldsymbol{x}^T L\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}} = \frac{\sum_{(v_i,v_j)\in\mathcal{E}} w_{ij}(x_i - x_j)^2}{\sum_{v_i\in\mathcal{V}} x_i^2}.$$

As L is real and symmetric, its eigendecomposition is given by $L = U\Lambda U^T$ where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_n), \lambda_1 \le \lambda_2 \le ... \le \lambda_n$, and $U = (\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_n)$ with $\boldsymbol{u}_i \in \mathbb{R}^n$ being the eigenvector for eigenvalue λ_i. We call $\tilde{\boldsymbol{x}} = U^T\boldsymbol{x}$ as the graph Fourier transform of the graph signal $\boldsymbol{x}$ and its inverse transform is given by $\boldsymbol{x} = U\tilde{\boldsymbol{x}}$.

GFT transfers the graph signal from the spatial domain to the spectral domain. Rayleigh quotient can be transformed into the spectral domain as follows:

$$Ray(\boldsymbol{x}) = \frac{\boldsymbol{x}^T L\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}} = \frac{\boldsymbol{x}^T U\Lambda U^T\boldsymbol{x}}{\boldsymbol{x}^T UU^T\boldsymbol{x}} = \frac{\tilde{\boldsymbol{x}}^T\Lambda\tilde{\boldsymbol{x}}}{\tilde{\boldsymbol{x}}^T\tilde{\boldsymbol{x}}} = \frac{\sum_{v_i\in\mathcal{V}} \lambda_i\tilde{x}_i^2}{\sum_{v_i\in\mathcal{V}} \tilde{x}_i^2}. \quad (1)$$

The graph filter $\mathcal{H}$ is defined as:

$$\mathcal{H} = Udiag(h(\lambda_1), h(\lambda_2)..., h(\lambda_n))U^T, \quad (2)$$

where $h(\cdot)$ is the frequency response function. The graph convolution of an input signal $\boldsymbol{x}$ and the filter $\mathcal{H}$ is defined as follows:

$$\boldsymbol{y} = \mathcal{H}\boldsymbol{x} = Udiag(h(\lambda_1), h(\lambda_2)..., h(\lambda_n))U^T\boldsymbol{x}.$$

In summary, the original graph signal $\boldsymbol{x}$ is first transformed into Fourier space by Fourier basis U^T, then the spectral domain signal is processed by filter $h(\cdot)$, and finally inversely transformed into spatial domain by U.

3.2 Motivation of Using GSP in CF

CF tasks aim to predict the real preference matrix according to the observed interaction matrix. If we have constructed the correct similarity matrix between items and the input signal is a user's real preference signal which indicates the user's preference for each item, the total variation of the graph signal can be discussed in two cases: 1) if two items i and j $(i \ne j)$ are similar, i.e., w_{ij} is large, the user's preferences for these two items will be similar, which means that $(x_i - x_j)^2$ should be small. This situation will not lead to excessive $TV(\boldsymbol{x})$; 2) if two items i and j $(i \ne j)$ are dissimilar, i.e., w_{ij} is small, the user often has different preferences for the two items, which means that $(x_i - x_j)^2$ should be large. This also does not cause $TV(\boldsymbol{x})$ to be too large because their similarity w_{ij} is small. In conclusion, if the real preference signal is used as input, the total variation should be small.

	1	2	3	4	5
1	0.8	0.9	0.4	0.1	0.2
2	0.7	0.8	0.8	0.9	0.1
3	0.1	0.2	0.8	0.9	0.3
4	0.8	0.3	0.8	0.1	0.9

(a) Real preference matrix

	1	2	3	4	5
1	0.8	0.9	0.4	0.1	x
2	x	0.8	0.8	0.9	0.1
3	0.1	x	x	0.9	0.3
4	x	0.3	0.8	0.1	0.9

(b) With exposure noise

	1	2	3	4	5
1	1	1	1	0	0
2	0	1	1	1	0
3	0	0	0	1	0
4	0	0	1	0	1

(c) With quantization noise

Figure 1: Relationship between the real preference matrix and observed preference matrix by considering noises.

However, due to the exposure noise and quantization noise in the observed interaction matrix [35], the total variation becomes larger when the input signal is the observed user interaction signal. Fig. 1(a) shows the real preference matrix, each row of which is a user's preference signal. Exposure noise means that users can not encounter all items, but only a subset of them. These items that are not exposed to users may or may not be liked by the users. Fig. 1(b) shows the observed change in the user preference matrix due to exposure noise. Users will choose whether to interact with the items exposed to them based on their preferences. Finally, our observed user-item ratings are binary variables, which are no longer the real preference values, due to quantization noise. Fig. 1(c) shows the change of user preference matrix observed by us due to quantization noise, that is, the final observed interaction matrix.

From the perspective of the spectral domain, the real preference signal is a low-frequency signal, and the two kinds of noise are high-frequency signals. The observed interaction signal is the fusion of low-frequency signal and high-frequency signal. We can effectively solve the CF problem by graph convolution, the key of which is to design a graph filter determined by the frequency response function. For Eq. (1), take $\boldsymbol{x} = \boldsymbol{u}_i$, we can get $Ray(\boldsymbol{u}_i) = \lambda_i$, indicating that the eigenvector corresponding to the small eigenvalue is smoother, and this smoothing is smooth in the sense of the whole graph, e.g., globally smooth. When restoring the spectral domain signal to the spatial domain, if only the smoother eigenvectors are used as the basis vectors, the reconstructed spatial signal will have a lower frequency in the sense of the whole graph.

Fig. 2(i) illustrates the item similarity graph used in previous works. we use 1 to represent strong node signal, and 0 to represent weak node signal. The input graph signal is $(1, 0, 0)$, which makes item 1 have a strong node signal, as shown in Fig. 2(ii). After graph convolution, the graph signal becomes smooth, that is, similar items have similar graph signals. This improves the signal of item 2 and then recommends it to the user a, as shown in Fig. 2(iii).

However, we believe that the real user preference and these two kinds of noise can not be simply separated from the spectral domain by frequency. Specifically, globally smooth signals are dominated by general preferences which may not be enough to capture

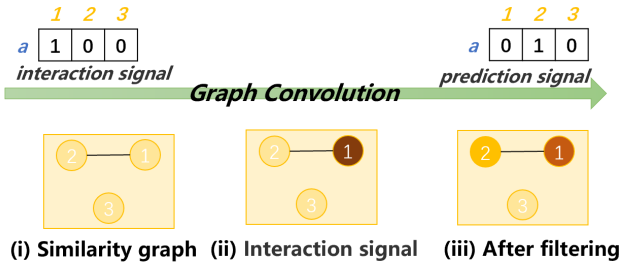

Figure 2: The previous GSP-based CF method. The interaction signal is directly input into the similarity graph.

personalized preferences. Therefore, it is necessary to introduce locally smooth but globally rough signals to capture personalized preferences, which can help to improve recommendation quality.

4 THE PGSP METHOD

In this section, we propose the personalized graph signal processing (PGSP) method which consists of three key components:

- *Personalized Graph Signal*, which has richer user-personalized information to describe users more accurately;
- *Augmented Similarity Graph*, which has more graph topology information to utilize the personalized graph signals more effectively;
- *Mixed-Frequency Graph Filter*, which allows globally smooth signals that reflect the general user preferences and locally smooth but globally rough signals that reflect the personalized user preferences to pass through.

We combine the above three components and propose the *PGSP Pipeline*, which consists of pre-processing, graph convolution and post-processing to achieve higher accuracy. In the following discussion, we suppose that there are m users and n items, and their interaction matrix is $R \in \{0,1\}^{m\times n}$, where $R_{ij} = 1$ indicates that user i has interacted with item j.

4.1 Personalized Graph Signal and Augmented Similarity Graph

In this section, we first introduce how to construct the similarity matrix when only relying on the graph structure from the perspective of random walk, and then propose personalized graph signal and augmented similarity graph.

4.1.1 Construction of Similarity Graph. When only considering the graph topology, we conclude from the design of *Node2Vec* [8] that the similarity of two nodes is determined by three factors: 1) the number of paths between two nodes, 2) the length of the paths and 3) the degree of the nodes that compose each of the paths. In random walk, there are usually a large number of short paths between two nodes with high similarity, and the degree of the nodes composing each of the paths should not be too large (otherwise the probability of walking through each path will be low).

For the CF problem with only user-item interaction information, all we have is the structure of the interaction graph, so these three factors should be taken into account as prior knowledge when constructing the similarity graph. The user-item interaction graph is a bipartite graph, which means that users are directly connected to items, and paths between two users/items must pass through intermediate items/users. Therefore, for user-user similarity, item-item similarity, and user-item similarity, we need to consider the degrees of user nodes and item nodes at the same time.

The interaction matrix indicates the existence of a path with a length of 1 between a user node and an item node. The original interaction matrix R can not be used to describe the similarity between users and items, because it does not consider the node degree. We take into account the degree of nodes and obtain the normalized interaction matrix, which can be used to represent the similarity between users and items as follows:

$$S_{UI} = (D_U)^{-1/2} R (D_I)^{-1/2}. \tag{3}$$

The D_U and D_I are diagonal matrix, where $(D_U)_{ii} = \sum_{j=1}^{n} R_{ij}, i = 1, ..., m$ and $(D_I)_{jj} = \sum_{i=1}^{m} R_{ij}, j = 1, ..., n$. From the perspective of random walk, the weight of an edge completely depends on the degree of the nodes connected by the edge. The more items that a user has interacted with, the less a single item can be capable of describing the user, and the lower the similarity between the user and the items. The same is true from the perspective of an item.

Based on the normalized interaction matrix, we can calculate the user-user similarity matrix and item-item similarity matrix, respectively, as follows:

$$S_U = S_{UI} S_{UI}^T, \quad S_I = S_{UI}^T S_{UI}. \tag{4}$$

From the perspective of random walk, the construction of S_U simulates the two-step walk process of transferring from the user nodes to the item nodes (represented by S_{UI}) and then to the user nodes (represented by S_{UI}^T). More paths with a length of 2 between two user nodes and higher weights of these paths will indicate higher similarity between the two user nodes. There is a similar conclusion for the item-item similarity matrix S_I.

We emphasize that co-occurrence relationship and cosine similarity are not suitable for calculating similarity in this scenario. In detail, their calculations are as follows:

$$S_U^{co} = RR^T, \quad S_U^{cos} = D_U^{-1/2} R R^T D_U^{-1/2}.$$
$$S_I^{co} = R^T R, \quad S_I^{cos} = D_I^{-1/2} R^T R D_I^{-1/2}.$$

Obviously, these two methods do not fully consider the influence of node degree on random walk.

4.1.2 Construction of Personalized Graph Signal. Due to data sparsity issues in many recommender systems, sparse historical interactions may not be enough to describe user preferences accurately. For each user, in addition to the direct interactions with items, there is also a collaborative relationship with other users, which is an indirect relationship. We combine these two kinds of information to describe users more accurately.

For each user, we concatenate his/her similarities with other users with his/her interaction signal to obtain the personalized graph signal as follows:

$$\tilde{R} = S_U || R, \tag{5}$$

where $||$ is the concatenation operation.

4.1.3 Construction of Augmented Similarity Graph. The interaction matrix is the direct embodiment of users' behaviors and describes the relationship between users and items. The similarity matrix is the compression of interaction information to describe the relationship between item-item and user-user. There are two motivations for the augmented similarity graph. Firstly, all these relationships are useful for GSP-based CF methods. Using the user-item relationship, the signals of user nodes can directly affect the signals of item nodes, and the existence of the item-item/user-user relationship makes it possible for the signals between item/user nodes to interact with each other. Secondly, the personalized graph signal is an $m+n$-dimensional vector, which requires a similarity graph that can accommodate more node signals.

We construct the augmented similarity graph, which contains all the above three relationships as follows:

$$A = \begin{bmatrix} S_U & S_{UI} \\ S_{UI}^T & S_I \end{bmatrix}. \tag{6}$$

Overall, our method is beneficial to CF in the following aspects: 1) a more reasonable similarity matrix constructed from the perspective of random walk is used to describe the relationship of users-items, users-users and items-items, 2) a personalized graph signal with richer information is used to describe users more accurately, and 3) an augmented similarity matrix with more topological information is used to process personalized graph signals more effectively.

4.2 Mixed-Frequency Graph Filter

In this section, we propose a mixed-frequency graph filter, which can capture globally smooth signals and locally smooth but globally rough signals simultaneously.

With the definition of the augmented similarity graph, we can define the corresponding Laplacian matrix $L = I - A$, where I is an identity matrix. The frequency response function of the ideal low-pass filter is $h(\lambda) = 1$ if $\lambda \leq \lambda_k$ and $h(\lambda) = 0$ otherwise, where λ_k is the cut-off frequency of the low-pass filter. Putting it into Eq. (2), we can obtain the ideal low-pass filter as follows:

$$\mathcal{H}_{ideal} = U diag(\overbrace{1, ..., 1}^{k}, \overbrace{0, ..., 0}^{m+n-k})U^T = U_k U_k^T,$$

where U is a matrix composed of the eigenvectors of L, U_k is the first k columns of U corresponding to the smaller eigenvalues. This indicates that the construction of the ideal low-pass filter does not need complete eigendecomposition of the matrix, but only needs to calculate the eigenvectors corresponding to the smaller eigenvalues.

An ideal low-pass filter only uses the low-frequency Fourier basis to reconstruct the spatial signal. Since these Fourier bases are low-frequency signals in the sense of the whole graph, it means that the signal filtered by an ideal low-pass filter is also a smooth signal in the sense of the whole graph. The left arrow in Fig. 3 describes this global smoothing. After filtering, the node signal spreads to the whole graph.

However, due to the sparsity of data, the interaction signal may not have enough information to restore the complete real preference through such global smoothing. On the contrary, only using the low-frequency signals will overdraft the expressive ability of the interaction signal for the filtered signals only containing general preference information, which damages the personalized information.

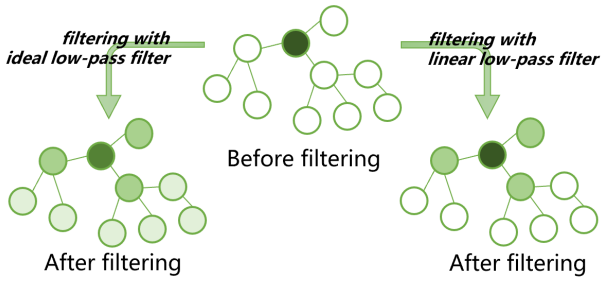

Figure 3: The ideal low-pass filter makes the signal globally smooth, and the linear low-pass filter makes the signal locally smooth but globally rough.

To solve the above problems, we implement a mixed-frequency graph filter, which is a low-pass filter. From the perspective of the spatial domain, the result of multiplication with augmented similarity matrix A is neighborhood aggregation, which leads to local smoothing. The right arrow in Fig. 3 describes this local smoothing, after filtering, the node signal only diffuses to the directly connected nodes. Theorem 1 shows that A is a linear low-pass filter with frequency response function $h(\lambda) = 1-\lambda$, which indicates that the filtered signal is not globally smooth because it allows some high-frequency components of the signal to pass through.

Theorem 1. *The augmented similarity matrix A is a low-pass filter with a frequency response function $h(\lambda) = 1 - \lambda$.*

Proof. First, we have:

$$L = U\Lambda U^T = U diag(\lambda_1, \lambda_2, ..., \lambda_{m+n})U^T,$$

which means that L is a graph filter with a frequency response function of $h(\lambda) = \lambda$.

Then, we have:

$$L = I - A.$$

Suppose $\boldsymbol{u}_i$ is the eigenvector of L corresponds to λ_i, the derivation is as follows:

$$L\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i = (I - A)\boldsymbol{u}_i = \boldsymbol{u}_i - A\boldsymbol{u}_i.$$

So, we have:

$$A\boldsymbol{u}_i = (1 - \lambda_i)\boldsymbol{u}_i,$$

which means that A and L have the same eigenvectors, and the corresponding eigenvalues have the relationship:

$$(\lambda_A)_i = 1 - \lambda_i,$$

where $(\lambda_A)_i$ is the i-th largest eigenvalue of A. Suppose:

$$\Lambda_A = diag((\lambda_A)_1, (\lambda_A)_2, ..., (\lambda_A)_{m+n}).$$

So, we have:

$$A = U\Lambda_A U^T = U diag(1 - \lambda_1, 1 - \lambda_2, ..., 1 - \lambda_{m+n})U^T,$$

which means that A is a low-pass filter with a frequency response function of $h(\lambda) = 1 - \lambda$. □

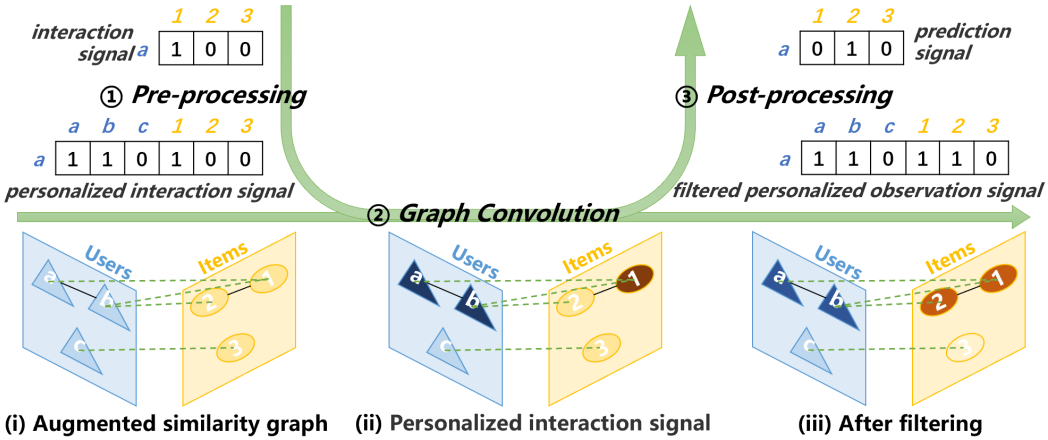

Figure 4: The proposed PGSP method. The personalized graph signal is constructed in the pre-processing step and input into the augmented similarity graph, which promotes users' potential interests.

To consider both general user preferences and personalized user preferences, we superimpose a linear low-pass filter that smooths signals locally on the basis of an ideal low-pass filter that smooths signals globally in a certain proportion as follows:

$$\mathcal{H} = (1-\phi)\mathcal{H}_{ideal} + \phi A, \tag{7}$$

where ϕ is a hyperparameter that controls the ratio of the globally smooth signal to the locally smooth signal.

4.3 PGSP Pipeline

In this section, we present how to combine the personalized graph signal, the augmented similarity graph and the mixed-frequency graph filter by proposing a pipeline named Personalized Graph Signal Processing which consists of pre-processing, graph convolution and post-processing.

1) *Pre-processing step.* We first obtain the personalized interaction matrix by Eq. (5). Then we normalize it to obtain the normalized personalized interaction matrix as follows:

$$\tilde{R}_{norm} = \tilde{R}\tilde{D}^{\beta}, \tag{8}$$

where $\tilde{D}_{jj} = \sum_{i=1}^{m} \tilde{R}_{ij}$ is a diagonal matrix, and $\beta \leq 0$ is a hyperparameter. Eq. (8) means that the more users an item interacts with, the more likely it is a popular item, and the lower ability to express users' personalized preferences.

2) *Graph convolution step.* We use the proposed mixed-frequency graph filter to filter the normalized personalized graph signals to get the filtered personalized graph signals:

$$\hat{\tilde{R}}_{norm} = \tilde{R}_{norm}\mathcal{H}. \tag{9}$$

3) *Post-processing step.* We restore the predicted personalized interaction matrix by multiplying $\tilde{D}^{-\beta}$, then the last n columns are intercepted as the predicted users' preferences matrix:

$$\hat{R} = (\hat{\tilde{R}}_{norm}\tilde{D}^{-\beta})_{:,-n:}, \tag{10}$$

where $:, -n:$ means to take all rows and last n columns of the matrix.

We illustrate how the PGSP method works through an example as shown in Fig. 4. For simplicity, we use 1 to represent a strong node signal, and 0 to represent a weak node signal. Suppose user a interacted with item 1, user b interacted with items 1 and 2, user c interacted with item 3, and user a is the target user. There is a strong similarity between the two connected nodes. User a is similar to user b because they both have interacted with item 1. Item 1 is similar to item 2 because they both have interacted with user b. We first construct an augmented similarity graph as shown in Fig. 4(i), which includes not only the user-user similarity (the blue part) and item-item similarity (the yellow part), but also the user-item interaction information (the dashed line). In pre-processing, the original interaction signal $(1, 0, 0)$ (indicating that user a has only interacted with item 1) concatenates the similarity signal $(1, 1, 0)$ (indicating that user a is similar to user b and himself/herself, but dissimilar to user c) and becomes $(1, 1, 0, 1, 0, 0)$ (user a's personalized graph signal). The personalized graph signal is input into the augmented similarity graph as shown in Fig. 4(ii). In graph convolution, with the proposed mixed-frequency graph filter, we obtain the predicted signal, which is a combination of globally smooth signals and locally smooth signals as shown in Fig. 4(iii). In this process, on one hand, on the item-item similarity graph, the strong signal of item 1 will be transmitted to item 2, causing the signal of item 2 to be strengthened. On the other hand, on the user-item similarity graph, user b who is similar to user a has a strong signal, which will promote the signal strength of item 2. User c who is dissimilar to user a has a weak signal, which will weaken the signal strength of item 3. Our method makes the items that users like have a higher intensity than those that users dislike. Finally, the predicted signal is $(0, 1, 0)$, which means that the model will recommend item 2, which may be the potential interests of user a because user b who is similar to user a has interacted with it, to user a.

Table 1: Statistics of the experimental data.

Dataset	#User	#Item	#Interaction	Density
Gowalla	29,858	40,981	1,027,370	0.084%
Yelp2018	31,668	38,048	1,561,406	0.130%
Amazon-Book	52,643	91,599	2,984,108	0.062%

Table 2: Performance comparison to state-of-the-art CF methods in recent two years. *RI* represents the relative improvement between PGSP and the corresponding method.

Dataset	Gowalla		Yelp2018		Amazon-book		Reported	Avg RI	
Method	recall	ndcg	recall	ndcg	recall	ndcg	by	recall	ndcg
LR-GCCF [3]	0.1519	0.1285	0.0561	0.0343	0.0335	0.0265	[19]	54.88%	71.75%
ENMF [2]	0.1523	0.1315	0.0624	0.0515	0.0359	0.0281	[19]	45.79%	47.70%
NIA-GCN [29]	0.1726	0.1358	0.0599	0.0491	0.0369	0.0287	[29] [28]	40.65%	46.80%
LightGCN [11]	0.1830	0.1554	0.0649	0.0530	0.0411	0.0315	[11]	28.95%	32.89%
DGCF [30]	0.1842	0.1561	0.0654	0.0534	0.0422	0.0324	[30]	26.94%	30.75%
NGAT4Rec [28]	-	-	0.0675	0.0554	0.0457	0.0358	[28]	30.27%	34.18%
SGL-ED [33]	-	-	0.0675	0.0555	0.0478	0.0379	[33]	26.86%	29.57%
DGCF [16]	0.1891	0.1602	0.0703	0.0575	0.0476	0.0369	[16]	17.16%	19.95%
MF-CCL [19]	0.1837	0.1493	0.0698	0.0572	0.0559	0.0447	[19]	11.01%	13.36%
SimpleX [19]	0.1872	0.1557	0.0701	0.0575	0.0583	0.0468	[19]	8.47%	9.75%
UltraGCN [20]	0.1862	0.1580	0.0683	0.0561	0.0681	0.0556	[20]	3.70%	3.51%
GF-CF [27]	0.1849	0.1518	0.0697	0.0571	0.0710	0.0584	[27]	1.83%	2.61%
PGSP (ours)	**0.1916**	**0.1605**	**0.0710**	**0.0583**	**0.0712**	**0.0587**		-	-

4.4 Comparison with GF-CF

As a GSP-based collaborative filtering method, GF-CF [27] is very similar to PGSP, and has achieved excellent results in recommendation tasks. However, GF-CF only uses user interaction signals as input signals. In addition, GF-CF classifies various collaborative filtering methods as filters with different frequency response functions, without analyzing the role of high-frequency components of observed signals. In contrast, PGSP depicts user interests more accurately through the personalized graph signal containing richer user information, and constructs an augmented similarity graph containing more graph topology information, to more effectively characterize user interests. We not only propose a mixed-frequency graph filter to introduce useful information in the high-frequency component of the observed signals by combining an ideal low-pass filter and a linear low-pass filter, but also conduct an experimental analysis on the influence of high-frequency information on recommendation accuracy.

5 EXPERIMENTS

We conduct extensive experiments to evaluate our method. Specifically, we evaluate the accuracy of the PGSP method, and analyze the effects of the personalized graph signal, augmented similarity graph, and mixed-frequency graph filter.

5.1 Experimental Settings

5.1.1 Dataset. We use three public datasets to evaluate the recommendation accuracy of our proposed method, including *Gowalla*, *Yelp2018* and *Amazon-Book*, released by the LightGCN work [11]. The statistics of the experimental data are shown in Table 1. For each dataset, we randomly select 80% of the historical interactions of each user to constitute the training set, and treat the remaining as the test set. From the training set, we randomly select 10% of interactions as the validation set to tune hyper-parameters.

5.1.2 Metrics. We use two popular evaluation protocols for top-N recommendation [11]: *Recall@K* and *NDCG@K* with $K = 20$.

5.1.3 Baselines. We compare the proposed PGSP method with three types of state-of-the-art methods in recent years as follows:

- MF-based methods, whose basic idea is to represent users and items in the same latent space with latent vectors, and use dot products as the strength of interaction possible between users and items, including ENMF [2], MF-CCL [19] and SimpleX [19].
- GNN-based methods, whose basic idea is to embed users and items in the space by means of message transmission. Each user and item has an embedding vector, and the dot product is used to represent the interaction possibility of different users and items. Including LR-GCCF [3], NIA-GCN [29], LightGCN [11], DGCF [30], NGAT4Rec [28], SGL-ED [33], UltraGCN [20] and DGCF [16].
- GSP-based GF-CF method [27], which sums up different collaborative filtering methods, including those based on the neighborhood, matrix factorization and graph neural network, into low-pass filters with different frequency response functions.

We only use these latest works for comparison, and the classic baselines are omitted due to space limitations.

5.2 Performance Comparison

Table 2 reports the performance comparison results and the relative improvement on each metric. Since the datasets partition and evaluation metrics of all compared baselines are consistent with our method, we directly present the results reported in them, we can ensure that we present the optimal performance of various methods on the premise of fair comparison. The experimental setting of all baselines can be found at https://openbenchmark.github.io/BARS [38]. We have the following observations from the results:

- Compared with SOTA methods in recent two years, our method has a significant improvement, especially in *NDCG*, which shows that our method not only optimizes the recall score, but also optimizes the ranking order.
- As a nonparametric method, PGSP is better than recently proposed parametric methods with a large margin, especially on *Amazon-Book*, a sparse dataset. This should be due to sparse data in CF, in which the data-driven methods may not work well.

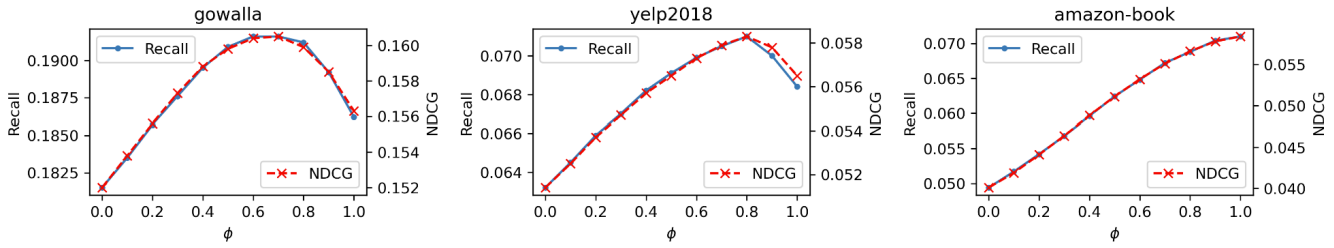

Figure 5: Effectiveness of high-frequency signals on three datasets. The x-axis is ϕ, which controls the weight of the high frequency (locally smooth but globally rough) information in the prediction signal. We can see that high-frequency signals can help to improve accuracy in all three datasets.

- Our method is superior to GF-CF, which is also based on GSP. This is because compared with GF-CF, our method uses the augmented similarity matrix and the personalized graph signal, and takes into account the role of high-frequency information and the normalized matrix.

5.3 Effectiveness of High-frequency Signal

We construct different mixed-frequency graph filters by controlling ϕ to change the proportion of globally smooth signals that reflect general user preferences and locally smooth but globally rough signals that reflect personalized user preferences. The results are shown in Fig. 5. We observe that the globally smooth signal filtered only by the ideal low-pass filter ($\phi = 0$) is not optimal. The accuracy can be improved by superimposing locally smooth signals filtered by the linear low-pass filter in proportion. Specifically, on *Gowalla* and *Yelp2018*, with the increase of locally smooth signals, the accuracy first increases and then decreases, and on *Amazon-Book*, the accuracy keeps increasing with the more locally smooth signal.

The density of the datasets is related to the relative performance increase obtained by adding high-frequency (locally smooth but globally rough) signals. As mentioned earlier, the expression ability of sparse interaction signals is weak, so we introduce high-frequency signals. According to this viewpoint, on more sparse datasets, we should observe higher performance improvement due to the introduction of high-frequency signals. In order to give a more convincing conclusion, we carried out a new experiment and added a dataset *ML-1M* [10]. The density of the datasets has such a relationship: *Amazon-Book* (0.062%) < *Gowalla* (0.084%) < *Yelp2018* (0.130%) < *ML-1M* (4.845%). Meanwhile, we conducted a 5-fold cross-validation and reported the average results to avoid the deviation caused by the dataset partition. We observed that compared with using only low-frequency signals, the relative improvements brought by high-frequency signals are as follows: *Amazon-Book* (81.25%) > *Gowalla* (17.89%) > *Yelp2018* (4.47%) > *ML-1M* (0.79%), which empirically confirmed our viewpoint. Since the recommender system datasets are sparse in most realistic scenarios, we believe the introduction of high-frequency signals is important.

5.4 Training Time Comparison

Different from the methods based on matrix factorization (MF) and graph neural network (GNN), PGSP is a nonparametric method, which does not require parameter learning, so it can achieve very high training efficiency. We compare PGSP with MF and GNN-based CF methods, and another GSP-based method GF-CF [27] in Table 3. It should be noted that the methods based on MF and GNN require multiple epochs to obtain a trained model. For ENMF and LightGCN, we show the training time for only one epoch, and they need to train multiple epochs until convergence. For PGSP and GF-CF, we show the total time required to complete the model training. It can be observed that the total time required to train PGSP is even less than the time to train one epoch in MF and GNN-based CF methods, which confirms the high efficiency of PGSP. Due to the introduction of additional information, PGSPrequires a slightly longer running time than the pure GSP-based GF-CF method.

Table 3: Training time comparison of three types of methods on the Gowalla dataset. In ENMF and LightGCN, we show the training time of one epoch, and they need multiple epochs during training. In PGSP and GF-CF, we show the total time required to complete model training.

Method	PGSP	ENMF	LightGCN	GF-CF
Running time	8m42s	11m5s	69m40s	5m31s
Need multiple epochs	✗	✓	✓	✗

6 CONCLUSION

In this work, we further promote the performance of GSP in CF tasks. We establish the similarity relationship from the perspective of random walk. Based on the proposed similarity, a personalized graph signal with more personalized information is proposed to describe users more accurately, and an augmented similarity graph with more topology information is constructed to utilize the signal more effectively. We reveal the importance of high-frequency signals in observed signals, and construct a mixed-frequency graph filter to simultaneously use the globally smooth signal and the locally smooth signal. Finally, the effectiveness of the proposed method is analyzed through comprehensive experiments on three public datasets.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants 62172106 and 61932007.

REFERENCES

[1] Chao Chen, Dongsheng Li, Junchi Yan, Hanchi Huang, and Xiaokang Yang. 2021. Scalable and explainable 1-bit matrix completion via graph signal learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 7011–7019.

[2] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient neural matrix factorization without sampling for recommendation. *ACM Transactions on Information Systems (TOIS)* 38, 2 (2020), 1–28.

[3] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2020. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 27–34.

[4] Siheng Chen, Aliaksei Sandryhaila, José MF Moura, and Jelena Kovačević. 2015. Signal recovery on graphs: Variation minimization. *IEEE Transactions on Signal Processing* 63, 17 (2015), 4609–4624.

[5] Fan RK Chung and Fan Chung Graham. 1997. *Spectral graph theory*. Number 92. American Mathematical Soc.

[6] Ganqu Cui, Jie Zhou, Cheng Yang, and Zhiyuan Liu. 2020. Adaptive graph encoder for attributed graph embedding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 976–985.

[7] Chenyuan Feng, Zuozhu Liu, Shaowei Lin, and Tony QS Quek. 2019. Attention-based graph convolutional network for recommendation system. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 7560–7564.

[8] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[9] Jiawei Han, Jian Pei, and Yiwen Yin. 2000. Mining frequent patterns without candidate generation. *ACM sigmod record* 29, 2 (2000), 1–12.

[10] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[11] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[12] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[13] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. 230–237.

[14] Weiyu Huang, Antonio G Marques, and Alejandro R Ribeiro. 2018. Rating prediction via graph signal processing. *IEEE Transactions on Signal Processing* 66, 19 (2018), 5066–5081.

[15] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[16] Haodong Liu, Bo Yang, and Dongsheng Li. 2021. Graph Collaborative Filtering Based on Dual-Message Propagation Mechanism. *IEEE Transactions on Cybernetics* (2021).

[17] Jiahao Liu, Dongsheng Li, Hansu Gu, Tun Lu, Peng Zhang, and Ning Gu. 2022. Parameter-free Dynamic Graph Embedding for Link Prediction. *arXiv preprint arXiv:2210.08189* (2022).

[18] Jeremy Ma, Weiyu Huang, Santiago Segarra, and Alejandro Ribeiro. 2016. Diffusion filtering of graph signals and its use in recommendation systems. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 4563–4567.

[19] Kelong Mao, Jieming Zhu, Jinpeng Wang, Quanyu Dai, Zhenhua Dong, Xi Xiao, and Xiuqiang He. 2021. SimpleX: A Simple and Strong Baseline for Collaborative Filtering. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1243–1252.

[20] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. 2021. UltraGCN: Ultra Simplification of Graph Convolutional Networks for Recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1253–1262.

[21] Sunil K Narang, Akshay Gadde, and Antonio Ortega. 2013. Signal processing techniques for interpolation in graph structured data. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 5445–5449.

[22] Sunil K Narang, Akshay Gadde, Eduard Sanou, and Antonio Ortega. 2013. Localized iterative methods for interpolation in graph structured data. In *2013 IEEE Global Conference on Signal and Information Processing*. IEEE, 491–494.

[23] Hoang Nt and Takanori Maehara. 2019. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550* (2019).

[24] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. 2018. Graph signal processing: Overview, challenges, and applications. *Proc. IEEE* 106, 5 (2018), 808–828.

[25] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2000. *Application of dimensionality reduction in recommender system-a case study*. Technical Report. Minnesota Univ Minneapolis Dept of Computer Science.

[26] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[27] Yifei Shen, Yongji Wu, Yao Zhang, Caihua Shan, Jun Zhang, B Khaled Letaief, and Dongsheng Li. 2021. How Powerful is Graph Convolution for Recommendation?. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1619–1629.

[28] Jinbo Song, Chao Chang, Fei Sun, Xinbo Song, and Peng Jiang. 2020. NGAT4Rec: Neighbor-Aware Graph Attention Network For Recommendation. *arXiv preprint arXiv:2010.12256* (2020).

[29] Jianing Sun, Yingxue Zhang, Wei Guo, Huifeng Guo, Ruiming Tang, Xiuqiang He, Chen Ma, and Mark Coates. 2020. Neighbor interaction aware graph convolution networks for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1289–1298.

[30] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 1001–1010.

[31] Xiaohan Wang, Pengfei Liu, and Yuantao Gu. 2015. Local-set-based graph signal reconstruction. *IEEE transactions on signal processing* 63, 9 (2015), 2432–2444.

[32] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*. PMLR, 6861–6871.

[33] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 726–735.

[34] Jiafeng Xia, Dongsheng Li, Hansu Gu, Jiahao Liu, Tun Lu, and Ning Gu. 2022. FIRE: Fast incremental recommendation with graph signal processing. In *Proceedings of the ACM Web Conference 2022*. 2360–2369.

[35] Wenhui Yu and Zheng Qin. 2020. Graph convolutional network for recommendation with low-pass collaborative filters. In *International Conference on Machine Learning*. PMLR, 10936–10945.

[36] HuaXin Zhang, Yu Liu, and KeYin Cao. 2019. Integrating Spectral-CF and FP-Growth for Recommendation. In *Proceedings of the 2019 2nd International Conference on E-Business, Information Management and Computer Science*. 1–7.

[37] Lei Zheng, Chun-Ta Lu, Fei Jiang, Jiawei Zhang, and Philip S Yu. 2018. Spectral collaborative filtering. In *Proceedings of the 12th ACM conference on recommender systems*. 311–319.

[38] Jieming Zhu, Quanyu Dai, Liangcai Su, Rong Ma, Jinyang Liu, Guohao Cai, Xi Xiao, and Rui Zhang. 2022. Bars: Towards open benchmarking for recommender systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2912–2923.