Performance Improvement of Recommendation Systems using User Evaluation Information

Heechan Kim[1], Yunsuk Kwak[1], Namgi Kim[2*]
[1]Student, Dept. of Computer Science, Kyonggi University, Suwon, Korea
[2]Professor, Dept. of Computer Science, Kyonggi University, Suwon, Korea

사용자 평가 정보를 활용한 추천 시스템의 성능 개선

김희찬[1], 곽윤석[1], 김남기[2*]
[1]경기대학교 컴퓨터과학과 학생, [2]경기대학교 컴퓨터과학과 교수

Abstract Existing Graph Signal Processing (GSP)-based recommender systems do not explicitly incorporate user ratings when constructing interaction graphs. We propose Rating-Aware Graph Filtering. The method independently filters interaction-based and rating-based adjacency matrices and combines their outputs through weighted aggregation. Experiments on ML-100K, ML-Latest-Small, ML-1M, and CiaoDVD evaluate F1, MRR, and NDCG using rating thresholds of 2 (Ratings2) and 5 (Ratings5). Compared with HiGSP, Ratings2 improves 23 of 24 dataset-metric comparisons, while Ratings5 improves all 24. However, Ratings2 slightly decreases MRR@10 on ML-1M. These results indicate that user ratings can improve recommendation performance, although gains depend on the dataset, rating threshold, combination weights, and evaluation metric.

Key Words : Recommendation system, Graph signal processing, Graph filtering, User rating, HiGSP

요 약 기존 그래프 신호 처리(Graph Signal Processing, GSP) 기반 추천시스템은 사용자와 아이템의 상호작용 여부를 중심으로 그래프를 구성하여 평점에 나타난 사용자 선호를 충분히 반영하지 못한다. 본 논문에서는 이러한 한계를 개선하기 위해 사용자 평점을 활용하는 Rating-Aware Graph Filtering 기법을 제안한다. 제안 기법은 일정 평점 이상인 상호작용으로 구성한 평점 기반 인접행렬과 기존 상호작용 기반 인접행렬에 각각 그래프 필터링을 수행하고, 결과를 가중 결합한다. ML-100K, ML-Latest-Small, ML-1M, CiaoDVD의 네 데이터셋에서 평점 임계값 2와 5를 적용한 Ratings2 및 Ratings5의 추천 성능을 F1, MRR, NDCG로 평가하였다. HiGSP 대비 Ratings2는 24개 데이터셋·지표 조합 중 23개에서, Ratings5는 24개 모두에서 성능이 향상되었으나, Ratings2의 ML-1M MRR@10은 소폭 감소하였다. 실험 결과는 사용자 평점의 활용이 추천 성능 개선에 기여하지만, 개선 효과는 데이터셋, 평점 임계값, 결합 가중치 및 평가 지표에 따라 달라짐을 보여준다.

주제어 : 추천시스템, 그래프 신호 처리, 그래프 필터링, 사용자 평점, HiGSP

This work was supported by Kyonggi University's Graduate Research Assistantship 2026. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP)-ICAN(ICT Challenge and Advanced Network of HRD) grant funded by the Korea government(Ministry of Science and ICT) (IITP-2024-00436954) and supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (RS-2020NR049579).
*교신저자 : 김남기(ngkim@kyonggi.ac.kr)
접수일 2026년 08월 06일 수정일 2026년 10월 01일 심사완료일 2026년 10월 16일

www.kci.go.kr

1. Introduction

With the rapid advancement of the Internet and the widespread adoption of mobile devices, the amount of available information and content has increased dramatically. In particular, as markets in various service domains such as e-commerce, streaming services, and social media continue to expand, the overwhelming abundance of products and content has made it increasingly difficult for users to identify items that best match their preferences. In response to this challenge, recommender systems have emerged as a technology that assists users in making appropriate choices. A recommender system is a type of Information Filtering technology that aims to improve user satisfaction by recommending information that users are likely to find interesting [1,2,3,4]. Therefore, recommender systems contribute to alleviating the problem of information overload while providing significant benefits to both users and service providers.

The two most important factors in recommender systems are training speed and recommendation accuracy. Training speed refers to how quickly a recommender system can learn from newly acquired data and provide users with appropriate recommendations in real time. Since user preferences and behavioral patterns change rapidly over time, a recommender system that fails to promptly reflect these changes may provide outdated recommendations, thereby reducing user satisfaction. In addition, recommendation accuracy is an important metric that indicates how relevant and appropriate the recommended items are for individual users. A highly accurate recommender system effectively captures user preferences and delivers personalized content [5,6,7,8,9,10].

To address these issues, this study proposes a novel method that improves the performance of a Graph Signal Processing (GSP)-based recommender system by incorporating user rating information. Conventional Graph Convolution Network (GCN)-based recommender systems effectively model complex graph structures through graph convolution operations. However, they require a large number of trainable parameters, and the optimization process involves substantial computational cost and long training time [11,12,13,14,15]. As a result, updating the recommender system model is time-consuming. To overcome this limitation, Graph Signal Processing-based recommender systems have been introduced as an efficient alternative for modeling relatively simple relationships [16,17,18,19,20,21].

Graph Signal Processing-based recommender systems perform recommendation through a low-pass filter, which consists of simple matrix operations and does not require the learning of trainable weight parameters. Consequently, the model can be updated efficiently with significantly lower computational overhead. Furthermore, their recommendation accuracy is comparable to or even higher than that of Graph Convolution Network-based recommender systems [18,19,20,21].

However, existing Graph Signal Processing-based recommender systems treat items with high user ratings and those with low user ratings equally without distinguishing between them when constructing the model [21]. Therefore, this study proposes a method that incorporates users' high rating information into the model to achieve more personalized recommendations.

2. Related Works

Recommender systems are generally classified into Content-Based Filtering, Collaborative Filtering, and Hybrid Filtering, which combines the strengths of both approaches [1,2,3,4].

Content-Based Filtering recommends items with characteristics similar to those previously preferred by the user by utilizing item attributes. Although it can generate recommendations without relying on other users' data, it is limited by the

www.kci.go.kr

difficulty of extracting item features and its inability to fully capture users' diverse interests.

Collaborative Filtering is based on the assumption that users with similar preferences tend to favor similar items [3,5,6,8,9,10,22]. It is categorized into User-Based and Item-Based Collaborative Filtering [7]. This approach does not require item feature information and can effectively discover users' potential interests. However, it suffers from the Cold Start Problem, the Data Sparsity Problem, and increasing computational complexity as user-item interactions grow.

Hybrid Filtering combines Content-Based and Collaborative Filtering to improve recommendation performance. It can alleviate the Cold Start Problem while enhancing recommendation diversity, although it introduces higher model complexity and computational cost.

With the advancement of deep learning-based Collaborative Filtering, Graph Convolution Networks (GCNs) have been widely applied to recommender systems [11]. Representative models include NGCF [12], LightGCN [13], SimGCL [14], and XSimGCL [15].

Graph Signal Processing (GSP) extends conventional signal processing to graph-structured data by exploiting graph topology [16,17]. Its core concepts include Graph Signals, the Graph Laplacian Matrix, Graph Filtering, and the Graph Fourier Transform (GFT). In particular, Graph Filtering transforms graph signals into the frequency domain, where Low-pass Filters suppress high-frequency components while preserving the similarities among neighboring nodes.

GF-CF (Graph Filter-based Collaborative Filtering) is a representative recommender system based on Graph Signal Processing [18]. To capture user-item similarities, GF-CF employs both Linear Low-pass and Ideal Low-pass Filters. Subsequently, several improved models were proposed, including PGSP, which incorporates richer user information [19], Turbo-CF, which improves computational efficiency [20], and HiGSP, which introduces hierarchical user clustering before graph filtering [21].

This study is based on HiGSP, which achieved state-of-the-art performance among Graph Signal Processing-based recommender systems as of 2024. HiGSP combines a cluster-based filtering module with a global-aware filtering module consisting of an Ideal Low-pass Filter and a High-order Low-pass Filter to effectively model user preferences and predict future interactions.

A previous study [18] compared the Graph Convolution Network-based model LightGCN with the Graph Signal Processing-based model GF-CF. Since GF-CF operates solely through matrix operations without trainable parameters, it achieves significantly higher computational efficiency while maintaining comparable or even superior recommendation performance on the Yelp2018 and Amazon-book datasets. Motivated by these advantages, this study investigates whether incorporating user rating information can improve the recommendation quality of a HiGSP-based framework.

3. The Proposed Scheme

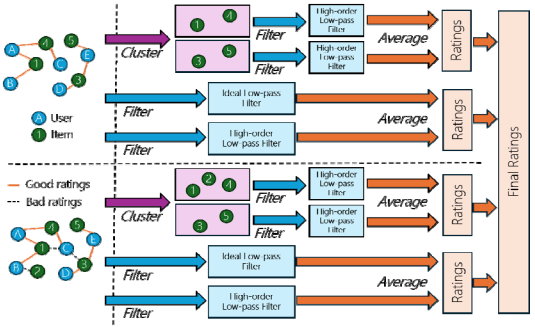
Existing Graph Signal Processing-based recommender systems, including HiGSP, construct the User Interaction Matrix for building the adjacency matrix by considering only whether an interaction exists between a user and an item, without taking user rating information into account [18,19,20,21]. This approach is based on the assumption that users interact more frequently with items they prefer. However, in real-world scenarios, users may repeatedly interact with items to which they assign low ratings, while interacting less frequently with items they rate highly. Consequently, an adjacency matrix constructed solely from interaction information may fail to adequately reflect users' actual

www.kci.go.kr

preferences, potentially leading to degraded recommendation performance.

To address this limitation, this study proposes a method that additionally incorporates user rating information during the construction of the User Interaction Matrix. Specifically, an observed user-item interaction is included in the rating-based interaction matrix only when its rating is greater than or equal to a threshold N. In the results reported in Section 4, N = 2 and N = 5 are denoted as Ratings2 and Ratings5, respectively. This approach aims to strengthen the signals of users' preferred items and improve personalized recommendation performance.

However, using only the rating-based adjacency matrix may result in information loss because interactions associated with low ratings are excluded. To overcome this issue, this study utilizes both the conventional interaction-based adjacency matrix and the proposed rating-based adjacency matrix. Graph filtering is independently performed on each adjacency matrix, and the resulting representations are combined using a weighted aggregation strategy. This approach effectively incorporates user preference information while minimizing information loss [16,17]. Furthermore, the proposed combination framework provides an ensemble effect. Section 4 compares the recommendation performance of the proposed method with that of HiGSP under rating thresholds of 2 and 5.



[Fig. 1] Architecture of the Proposed Model

Figure 1 illustrates the overall architecture of the proposed model. The framework consists of two main components. The lower branch performs graph filtering in the same manner as the original HiGSP model, while the upper branch performs graph filtering using the newly constructed adjacency matrix proposed in this study [21]. By combining these two branches, the proposed model alleviates the information loss problem while achieving more personalized recommendations and benefiting from an ensemble effect.

The lower branch of Figure 1 constructs the adjacency matrix using the same procedure as the original HiGSP model to prevent information loss caused by the proposed method. After constructing the adjacency matrix, clustering is performed to group similar users, followed by the application of a High-order Low-pass Filter to compute the rating scores. Subsequently, the original adjacency matrix is processed using an Ideal Low-pass Filter and a High-order Low-pass Filter, respectively, to generate additional rating scores [21]. Through the application of multiple Low-pass Filters, the model effectively captures user preferences and improves recommendation quality.

The upper branch of Figure 1 constructs a new adjacency matrix using the proposed method and performs graph filtering based on this newly generated graph structure. Similar to the lower branch, clustering is first applied to group users with similar preferences, followed by a High-order Low-pass Filter to compute rating scores. The proposed adjacency matrix is then processed using both an Ideal Low-pass Filter and a High-order Low-pass Filter to generate rating scores. Unlike conventional approaches that rely solely on interaction information, the proposed method incorporates users' high rating information into the graph filtering process, enabling more accurate and personalized recommendations. In the final stage, the four

www.kci.go.kr

rating matrices generated from the above procedures are combined using weighted aggregation to produce the final rating matrix. Through this framework, the proposed model addresses the limitations of the original HiGSP while simultaneously benefiting from an ensemble effect, thereby improving recommendation performance.

The proposed method introduces additional computation compared with HiGSP because the interaction-based and rating-based adjacency matrices are processed separately. The additional branch requires construction of a rating-based graph, user clustering, and low-pass filtering, followed by weighted aggregation of the resulting rating matrices. The reported recommendation improvements should therefore be considered together with this additional computational burden. The experiments in this paper evaluate recommendation quality but do not measure execution time or memory consumption. Consequently, the magnitude of the computational overhead and its effect on overall efficiency cannot be quantified from the reported results.

4. Experiments

Table 1 summarizes the four datasets for which results are reported in this paper: ML-100K, ML-Latest-Small, ML-1M, and CiaoDVD. The MovieLens datasets contain user-movie ratings and are provided by GroupLens Research [23]. CiaoDVD contains product reviews and ratings [24]. For ML-100K and ML-1M, the training and test sets were newly partitioned for the present experiments. The table reports the numbers of users, items, and observed interactions, together with the rating ranges and interaction densities. Interaction density represents the percentage of all possible user-item pairs for which an interaction is observed.

〈Table 1〉 Datasets Used in the Experiments

Dataset	ML-100K	ML-Latest-Small	ML-1M	Ciao DVD
# of Relations	100K	100K	1M	72K
# of Users	943	610	6,040	17,615
# of Items	1,682	9,724	3,706	16,121
Rating Range	[1, 5]	[0.5, 5.0]	[1, 5]	[1, 5]
Density (%)	6.305	1.700	4.468	0.026

〈Table 2〉 Experimental Environment

Resource	Property
OS	Ubuntu 22.04 LTS
CPU	13th Gen Intel(R) Core TM I9-13990K
RAM	DDR 5 32GB + 256G SWAP
GPU	NVIDIA GeForce RTX 4090

The experiments in this study were conducted under the experimental environment described in Table 2 using the package versions listed in Table 3.

〈Table 3〉 Package Versions Used in the Experimental Environment

Package	Version
Python	3.7.10
Numpy	1.19.5
Pytorch	1.9.0
Scikit-learn	0.22.0
Scipy	1.5.2
Sparsesvd	0.2.2

〈Table 4〉 Hyperparameters Used in the Experiments

Hyperparameter	Setting
alpha1	0.3
alpha2	0.7
order1	8
order2	8
pri_factor	256
n_cluster	5
HiGSP : Ours	5 : 5 (default)
random_state	2024

The combination ratio is 5:5 by default. The dataset-specific exceptions for Ratings5 are described with Table 6.

www.kci.go.kr

Table 4 lists the hyperparameters used in the experiments. The parameters alpha1 and alpha2 control the contributions of the Ideal Low-pass Filter and High-order Low-pass Filter outputs, respectively. The parameters order1 and order2 specify the filter orders for the cluster-level and global-level operations, respectively. The parameter pri_factor determines the number of singular components used by sparsesvd, and n_cluster specifies the number of user clusters. The random_state parameter was fixed at 2024.

Recommendation performance is evaluated using F1@5, F1@10, MRR@5, MRR@10, NDCG@5, and NDCG@10. The results reported in Tables 5 and 6 correspond to Ratings2 and Ratings5, which retain observed interactions with ratings greater than or equal to 2 and 5, respectively.

〈Table 5〉 Performance Comparison for Ratings2 (Rating ≥ 2)

Dataset	Metrics	HiGSP	Ours(≥ 2)
ML-100K	F1@5	0.2099	**0.2139 (+1.91%)**
	F1@10	0.2718	**0.2750 (+1.18%)**
	MRR@5	0.5830	**0.5971 (+2.42%)**
	MRR@10	0.5434	**0.5511 (+1.42%)**
	NDCG@5	0.6366	**0.6484 (+1.85%)**
	NDCG@10	0.6182	**0.6226 (+0.71%)**
ML-Latest-Small	F1@5	0.1001	**0.1050 (+4.90%)**
	F1@10	0.1478	**0.1501 (+1.56%)**
	MRR@5	0.3270	**0.3495 (+6.88%)**
	MRR@10	0.3421	**0.3520 (+2.89%)**
	NDCG@5	0.3977	**0.4188 (+5.31%)**
	NDCG@10	0.4439	**0.4523 (+1.89%)**
ML-1M	F1@5	0.1789	**0.1805 (+0.89%)**
	F1@10	0.2457	**0.2462 (+0.20%)**
	MRR@5	0.5984	**0.6000 (+0.27%)**
	MRR@10	**0.5603**	0.5601 (−0.04%)
	NDCG@5	0.6571	**0.6591 (+0.30%)**
	NDCG@10	0.6407	**0.6410 (+0.05%)**
Ciao DVD	F1@5	0.0080	**0.0083 (+3.75%)**
	F1@10	0.0085	**0.0092 (+8.24%)**
	MRR@5	0.0133	**0.0139 (+4.51%)**
	MRR@10	0.0136	**0.0151 (+11.03%)**
	NDCG@5	0.0166	**0.0170 (+2.41%)**
	NDCG@10	0.0204	**0.0223 (+9.31%)**

Table 5 compares HiGSP with Ratings2, in which observed interactions with ratings greater than or equal to 2 are retained in the additional rating-based graph. Ratings2 improves 23 of the 24 reported dataset-metric comparisons. On ML-Latest-Small, the relative improvements reach 4.90% in F1@5, 6.88% in MRR@5, and 5.31% in NDCG@5. In contrast, the improvements on ML-1M are relatively small. In particular, MRR@10 decreases from 0.5603 to 0.5601, corresponding to an absolute change of −0.0002 and a relative change of approximately −0.04%. Thus, incorporating rating information does not improve every evaluation metric uniformly.

A possible explanation for the mixed result on ML-1M is that thresholding changes the information supplied by the additional rating-based branch, and combining this information with the interaction-based predictions may alter the ordering of recommended items. Such changes need not benefit all ranking metrics simultaneously. However, the reported results alone do not establish the cause of the decrease or its statistical significance. The decrease should therefore be acknowledged as a limitation of the observed improvement rather than dismissed as random variation.

〈Table 6〉 Performance Comparison for Ratings5 (Rating ≥ 5)

Dataset	Metrics	HiGSP	Ours(≥ 5)
ML-100K	F1@5	0.2099	**0.2115 (+0.76%)**
	F1@10	0.2718	**0.2728 (+0.37%)**
	MRR@5	0.5830	**0.5902 (+1.23%)**
	MRR@10	0.5434	**0.5490 (+1.03%)**
	NDCG@5	0.6366	**0.6434 (+1.07%)**
	NDCG@10	0.6182	**0.6219 (+0.60%)**
ML-Latest-Small	F1@5	0.1001	**0.1080 (+7.89%)**
	F1@10	0.1478	**0.1491 (+0.88%)**
	MRR@5	0.3270	**0.3490 (+6.73%)**
	MRR@10	0.3421	**0.3429 (+0.23%)**
	NDCG@5	0.3977	**0.4195 (+5.48%)**
	NDCG@10	0.4439	**0.4480 (+0.92%)**

www.kci.go.kr

ML-1M	F1@5	0.1789	**0.1804 (+0.84%)**
	F1@10	0.2457	**0.2473 (+0.65%)**
	MRR@5	0.5984	**0.6014 (+0.50%)**
	MRR@10	0.5603	**0.5625 (+0.39%)**
	NDCG@5	0.6571	**0.6607 (+0.55%)**
	NDCG@10	0.6407	**0.6427 (+0.31%)**
Ciao DVD	F1@5	0.0080	**0.0092 (+15.00%)**
	F1@10	0.0085	**0.0089 (+4.71%)**
	MRR@5	0.0133	**0.0155 (+16.54%)**
	MRR@10	0.0136	**0.0141 (+3.68%)**
	NDCG@5	0.0166	**0.0190 (+14.46%)**
	NDCG@10	0.0204	**0.0212 (+3.92%)**

Table 6 presents the results for Ratings5, which retains observed interactions with ratings greater than or equal to 5 in the additional rating-based graph. Ratings5 improves all 24 reported dataset-metric comparisons with HiGSP. On ML-Latest-Small, F1@5 increases from 0.1001 to 0.1080, corresponding to a relative improvement of 7.89%. For the Ratings5 results on ML-Latest-Small, the HiGSP and rating-based predictions are combined with weights of 0.7 and 0.3, respectively. The corresponding weights for ML-1M are 0.8 and 0.2. These dataset-specific combination weights should be considered when interpreting differences between Ratings2 and Ratings5; the observed differences should not be attributed solely to the rating threshold.

CiaoDVD exhibits a clear difference between the two rating thresholds across recommendation cutoffs. At K = 5, Ratings5 yields higher scores than Ratings2 for all three metrics, achieving relative improvements over HiGSP of 15.00% in F1@5, 16.54% in MRR@5, and 14.46% in NDCG@5. At K = 10, however, Ratings2 yields higher scores than Ratings5, with relative improvements of 8.24% in F1@10, 11.03% in MRR@10, and 9.31% in NDCG@10. Therefore, neither threshold is uniformly superior across all evaluation metrics, and the preferred setting depends on the recommendation cutoff and evaluation objective.

According to Table 1, CiaoDVD has the lowest interaction density among the reported datasets, at approximately 0.026%. One possible interpretation is that the rating-based branch provides a more selective preference signal, while the interaction-based branch retains the broader interaction structure. However, the present results do not establish that sparsity is the cause of the improvement. The large relative gains must also be interpreted in the context of the low baseline scores. For example, the 16.54% improvement in MRR@5 for Ratings5 corresponds to an absolute increase from 0.0133 to 0.0155, or 0.0022. The relative percentage gains and the absolute score differences should therefore be considered together when interpreting the results on CiaoDVD.

5. Conclusions

In this study, we proposed a method to improve the recommendation performance of Graph Signal Processing-based recommender systems by addressing the limitations of the existing HiGSP model [21]. In the original HiGSP model, user-item interactions associated with both high and low user ratings are treated equally when constructing the adjacency matrix, which limits the ability to generate personalized recommendations. To overcome this limitation, we introduced an additional adjacency matrix that incorporates items receiving high user ratings. This approach strengthens the signals of users' preferred items while providing an ensemble effect, thereby enabling more accurate prediction of user preferences.

This paper reports the performance of Ratings2 and Ratings5 on four benchmark datasets: ML-100K, ML-Latest-Small, ML-1M, and CiaoDVD. Compared with HiGSP, Ratings2 improves 23 of the 24 reported dataset-metric comparisons, while Ratings5 improves all 24 comparisons. Nevertheless, Ratings2 exhibits a slight decrease in MRR@10 on ML-1M. On CiaoDVD, Ratings5 achieves larger improvements at K = 5, whereas Ratings2 achieves larger

www.kci.go.kr

improvements at K = 10. These findings indicate that incorporating rating information can improve recommendation quality, but the benefit depends on the dataset, rating threshold, combination weights, and evaluation metric.

This study builds upon our previous conference work [25] by further examining the rating-aware graph filtering approach for recommendation systems. Although the proposed method improves recommendation performance in several evaluation metrics, its use of two adjacency matrices introduces additional computational overhead compared with the original HiGSP model. In particular, independently processing the interaction-based and rating-based adjacency matrices requires additional graph construction, filtering, and aggregation operations. This represents a trade-off between recommendation accuracy and computational efficiency. Because execution time and memory consumption were not measured in the reported experiments, the magnitude of this trade-off remains an open question. As future work, we plan to investigate a unified rating-weighted adjacency matrix that incorporates both interaction information and user preferences into a single graph structure. This approach aims to reduce redundant filtering operations while preserving the recommendation performance achieved by the proposed method.

REFERENCES

[1] A.I.Saleh, A.I.El Desouky and S.H.Ali, "Promoting the performance of vertical recommendation systems by applying new classification techniques," Knowledge-Based Systems, Vol.75, pp.192-223, 2015.

[2] M.Goyani and N.Chaurasiya, "A review of movie recommendation system: Limitations, Survey and Challenges," ELCVIA: Electronic Letters on Computer Vision and Image Analysis, Vol.19, No.3, pp.18-37, 2020.

[3] Y.Hu, Y.Koren and C.Volinsky, "Collaborative Filtering for Implicit Feedback Datasets," 2008 Eighth IEEE International Conference on Data Mining, pp.263-272, 2008.

[4] E.Çano and M.Morisio, "Hybrid Recommender Systems: A Systematic Literature Review," Intelligent Data Analysis, Vol.21, No.6, pp.1487-1524, 2017.

[5] D.Goldberg, D.Nichols, B.M.Oki and D.Terry, "Using Collaborative Filtering to Weave an Information Tapestry," Communications of the ACM, Vol.35, No.12, pp.61-70, 1992.

[6] P.Resnick, N.Iacovou, M.Suchak, P.Bergstrom and J.Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, pp.175-186, 1994.

[7] B.Sarwar, G.Karypis, J.Konstan and J.Riedl, "Item-based Collaborative Filtering Recommendation Algorithms," Proceedings of the 10th International Conference on World Wide Web, pp.285-295, 2001.

[8] Y.Koren, R.Bell and C.Volinsky, "Matrix Factorization Techniques for Recommender Systems," Computer, Vol.42, No.8, pp.30-37, 2009.

[9] P.Covington, J.Adams and E.Sargin, "Deep Neural Networks for YouTube Recommendations," Proceedings of the 10th ACM Conference on Recommender Systems, pp.191-198, 2016.

[10] X.He, L.Liao, H.Zhang, L.Nie, X.Hu and T.Chua, "Neural Collaborative Filtering," Proceedings of the 26th International Conference on World Wide Web, pp.173-182, 2017.

[11] T.N.Kipf and M.Welling, "Semi-supervised Classification with Graph Convolutional Networks," Proceedings of the International Conference on Learning Representations (ICLR), 2017.

[12] X.Wang, X.He, M.Wang, F.Feng and T.S.Chua, "Neural Graph Collaborative Filtering," Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.165-174, 2019.

[13] X.He, K.Deng, X.Wang, Y.Li, Y.Zhang and M.Wang, "LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation," Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.639-648, 2020.

[14] J.Yu, H.Yin, X.Xia, T.Chen, L.Cui and Q.V.H.Nguyen, "Are Graph Augmentations Necessary? Simple Graph Contrastive Learning for Recommendation," Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.1294-1303, 2022.

[15] J.Yu, X.Xia, T.Chen, L.Cui, N.Q.V.Hung and H.Yin, "XSimGCL: Towards Extremely Simple Graph Contrastive Learning for Recommendation," IEEE Transactions on Knowledge and Data Engineering, Vol.36, No.2, pp.913-926, 2024.

[16] X.Dong, D.Thanou, L.Toni, M.Bronstein and P.Frossard, "Graph Signal Processing for Machine Learning: A

www.kci.go.kr

Review and New Perspectives," arXiv:2007.16061, 2020.

[17] R.Ramakrishna, H.T.Wai and A.Scaglione, "A User Guide to Low-Pass Graph Signal Processing and Its Applications: Tools and Applications," IEEE Signal Processing Magazine, Vol.37, No.6, pp.74-85, 2020.

[18] Y.Shen, Y.Wu, Y.Zhang, C.Shan, J.Zhang, B.K.Letaief and D.Li, "How Powerful is Graph Convolution for Recommendation?," Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp.1619-1629, 2021.

[19] J.Liu, D.Li, H.Gu, T.Lu, P.Zhang, L.Shang and N.Gu, "Personalized Graph Signal Processing for Collaborative Filtering," Proceedings of the ACM Web Conference 2023, pp.1264-1272, 2023.

[20] J.-D.Park, Y.-M.Shin and W.-Y.Shin, "Turbo-CF: Matrix Decomposition-Free Graph Filtering for Fast Recommendation," Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.2672-2676, 2024.

[21] J.Xia, D.Li, H.Gu, T.Lu, P.Zhang, L.Shang and N.Gu, "Hierarchical Graph Signal Processing for Collaborative Filtering," Proceedings of the ACM Web Conference 2024, pp.3229-3240, 2024.

[22] M.Naumov, D.Mudigere, H.J.M.Shi, J.Huang, N.Sundaraman, J.Park, X.Wang, U.Gupta, C.J.Wu, A.G.Azzolini, D.Dzhulgakov, A.Mallevich, I.Cherniavskii, Y.Lu, R.Krishnamoorthi, A.Yu, V.Kondratenko, S.Pereira, X.Chen, W.Chen, V.Rao, B.Jia, L.Xiong and M.Smelyanskiy, "Deep Learning Recommendation Model for Personalization and Recommendation Systems," arXiv:1906.00091, 2019.

[23] F.M.Harper and J.A.Konstan, "The MovieLens Datasets: History and Context," ACM Transactions on Interactive Intelligent Systems, Vol.5, No.4, Article 19, pp.1-19, 2015.

[24] G.Guo, J.Zhang, D.Thalmann and N.Yorke-Smith, "ETAF: An Extended Trust Antecedents Framework for Trust Prediction," Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp.540-547, 2014.

[25] H.-C.Kim, J.Jeon, K.Chung and N.Kim, "Effects of User Ratings in GSP-Based Recommendation Systems," Proceedings of the 2025 Sixteenth International Conference on Ubiquitous and Future Networks (ICUFN), pp.513-515, 2025.
doi: 10.1109/ICUFN65838.2025.11169854.

김 희 찬(Heechan Kim) [준회원]



- 2023년 2월 : 경기대학교 AI컴퓨터공학부(공학사)
- 2025년 2월 : 경기대학교 컴퓨터과학과 석사

〈관심분야〉
강화학습, 메타러닝, 네트워크

곽 윤 석(Yunsuk Kwak) [준회원]



- 2026년 2월 : 경기대학교 인공지능전공 학사
- 2026년 3월 ~ 현재 : 경기대학교 컴퓨터과학과 석사과정

〈관심분야〉
인공지능, 컴퓨터 비전, 딥러닝, 개인정보 보호

김 남 기(Namgi Kim) [정회원]



- 1997년 : 서강대학교 컴퓨터공학과 학사
- 2000년 : KAIST 전산학과 석사
- 2005년 : KAIST 전산학과 박사
- 2007년 ~ 현재 : 경기대학교 AI컴퓨터공학부 교수

〈관심분야〉
컴퓨터 네트워크, 이상 탐지, 추천시스템 최적화, 멀티캐스트 라우팅

www.kci.go.kr