IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received August 16, 2020, accepted September 2, 2020, date of publication September 28, 2020, date of current version October 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3027380

A Hybrid Recommender System for Sequential Recommendation: Combining Similarity Models With Markov Chains

YEONGWOOK YANG[1], HONG-JUN JANG[1], AND BYOUNGWOOK KIM[2]

[1]Department of Computer Science and Engineering, Korea University, Seoul 02841, South Korea
[2]Department of Computer Engineering, Dongguk University, Gyeongju 38066, South Korea

Corresponding author: Byoungwook Kim (bwkim@dongguk.edu)

This work was supported in part by the Dongguk University Research Fund of 2018, and in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2019R1F1A1064019.

ABSTRACT With the growth of artificial intelligence technology, the importance of recommender systems that recommend personalized content has increased. The general form of the recommender system usually analyzes the users' log information to provide them with contents that they are interested in. However, to enable users to receive more suitable and personalized content, additional information must be considered besides the user's log information. We develop, in the present study, a hybrid recommender system that unifies similarity models—collaborative and content-based—with Markov chains for a sequential recommendation (called U2CMS). U2CMS takes into account both sequential patterns and information about contents to find accurate relationships between items. It uses a higher-order Markov chain to model sequential patterns over several time steps, as well as the textual information of the content to model the recommender system. To show the effectiveness of the U2CMS—with regard to handling sparsity issues, different N-ordered Markov Chain, and accurately identifying similarities between items, we carried out several experiments on various Amazon datasets. Our results show that the U2CMS not only has superior performance compared to existing state-of-the-art recommendation systems (including deep-learning based systems), but also it successfully handles sparsity issues better than other approaches. Moreover, U2CMS appears to perform stable when it comes to different N-ordered Markov Chain. Lastly, through visualization, we show the success of our proposed content-based filtering model in identifying similar items.

INDEX TERMS Item similarity model, content-based filtering, hybrid recommendation, sequential recommendation.

I. INTRODUCTION

The importance of recommender systems that recommend personalized contents has considerably increased during the past decade due to the rapid advancement in artificial intelligence technology. Recommender systems usually analyze activity data of users in systems to offer content that is of interest to them. Nonetheless, the recommended content to the users might not be of high quality or sufficient as some important available information of the users are not used in the recommendation process. To address this issue, and have improved recommended contents, additional information must be considered besides adding the user's log information. Considering both user's data and information enables the recommender system to find the content of the user's interest easier. In other words, this additional information helps to provide a more accurate answer to the question of "What kind of content does this user usually prefer?". This method called content-based filtering which uses information about content and profile of the user [1]. The core idea in this method is to consider the relationship between items based on information about the content. However, while this method may take into account only the general preference of the user, it is difficult to recommend a product that is likely to match the recently purchased products because it does not consider sequential information. Sequential information therefore can be the answer to this question.

In general, the sequential information is modeled by Markov chains to capture sequential patterns. Rendle *et al.* [2], for instance, proposed a factorized personalized Markov

The associate editor coordinating the review of this manuscript and approving it for publication was Zeshui Xu.

chains called FPMC that outperforms traditional recommendation models by combining sequential information with user preference information. Despite the successful results of the FPMC, this model considers only the previous order information of the user, and the calculation is complicated due to the separation of the sequential pattern matrix and the user preference matrix. Similarly, He and McAuley [3] presented a factorized sequential prediction with item similarity model called Fossil which includes the advantages of the aforementioned approach. Besides, it adds the high-order Markov chains concept, and considers the item similarity model approach proposed by Kabbur *et al.* [4]. Although Fossil surpassed the other previous methods, it only relies on the user's log information, so there is a limit to identifying similarity between items. More details on comparison between the existing recommender models and our proposed model is given in Section V (Table. 3).

In this paper, to address these gaps, we develop a hybrid approach inspired by the Fossil and content-based Filtering approaches [5]–[8], which combines the similarity-based approach with the high-order Markov chain and the similarity between items (called U2CMS). Our model takes into account both the sequential pattern and the information about the item to have a richer understanding of existing relationships between items. In brief, it firstly employs a novel content-based filtering approach based on the item similarity model to find similarity between the set of items and user's items. To do so, it uses a term frequency-inverse document frequency (TF-IDF) method, which uses the textual information of the content, to possibly include more useful information about the item. Secondly, it applies an item similarity-based method to recognize user's long-term preference (to model user preference). Finally, to find the short-term preference of a user, it uses high-order Markov chain to have more meaningful interactions in the user's item sequence. In other words, our model integrates three different methods that includes user preference, sequential activity, and the content-based approach with item similarity models. This enables the U2CMS to successfully reduce the dimension of the final model, decreasing computation time and complexity. Moreover, as the U2CMS emphasizes the relationship between items it can handle data sparsity [9]–[11], and it outperforms existing advanced recommender systems.

In Fig. 1, we present an example of the U2CMS recommendation in order to more clearly show the capacity of our proposed approach. If a specific user previously watched some fantasy movies (Beauty and Beast, Aladdin and Maleficent 1), the U2CMS recommends Maleficent 2. This result is produced by user preference, sequential patterns and the similarity of content. The user preference reflects that Maleficent 2 is similar to movies already watched (i.e., fantasy movies). The sequential patterns discover that a recommended movie frequently follows the recently watched movie Maleficent 1. The content similarity is then used to recommend the movie from a movie list which has movies with similar content.

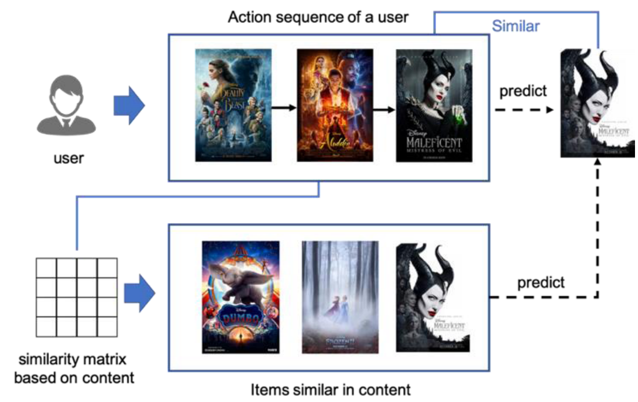


FIGURE 1. An example of U2CMS recommendation.

Contributions of this study are as follows:

- Proposing a new content-based filtering approach based on the item similarity model.
- Unifying content-based filtering and user preference model, and high-order Markov chains with factorized-based similarity models on sequential data environments.
- Showing the importance of including item's information and considering extended sequential patterns in alleviating the data sparsity problem in recommender systems.
- Showing the superiority of our proposed model to a variety of advanced existing approaches on real datasets of various sizes and types.

II. RELATED WORKS

A. COLLABORATIVE FILTERING

A recommender system generally uses collaborative filtering (CF) to predict a recommendation list through explicit feedback information, such as star ratings or reviews.

There exists different versions of CF recommender systems, for instance, those that are neighbor-based and use k-nearest neighbors algorithm for a given user or product based on a given similarity [12]–[14]. To calculate the similarity, this method can use the various similarity measures such as Jaccard similarity, cosine similarity and Pearson correlation coefficient. If a specific user and other users show similar purchasing patterns, their similarity scores will be high. If they are not similar, their similarity scores will be low.

Model-based CF uses machine learning models such as the Bayesian method [15], [16] or the matrix factorization (MF) method [17]–[22]. For example, Pan and Scholz [23] used the MF mechanism that uses a user and an item latent factor. For training their model, they employed hinge and least-squares optimization. Hu *et al.* [24], similarly, to learn the recommendation model and factorize a user/item matrix, used least-squares optimization.

Recently, sparse linear method (SLIM) [25] was proposed using an item-item similarity matrix to handle implicit feedback data (such as purchases and clicks), which outperforms the general recommendation model. Kabbur *et al.* [4] also proposed the factored item similarity model (FISM) that handles data sparsity by applying the MF mechanism to the item-item similarity matrix. However, above-mentioned approaches mostly ignore sequential patterns which are a crucial factor in generating more suitable recommended content.

B. CONTENT-BASED FILTERING

Content-based filtering [1], [26], [27] suggests items that are similar to those previously liked by users. This method takes account of user profile information (such as user preferences), as well as item's information (such as title, description). A common approach is to represent both users and items in the same vector space, and calculate user-item similarity scores between them. Similarity scores are calculated by a similarity method (e.g., Cosine similarity and Jaccard similarity).

Salton and McGill [28] proposed a web-page recommender system that uses the user queries information. Similarly, a book recommender system using text classification techniques is presented by Mooney and Roy [26]. Robin and Maarten [27] proposed an article recommender system, which is called PRES, using the text information, whereas Billsus and Pazzani [1] proposed a news recommender system, which uses the similarity between news. However, they are mostly vulnerable to the data sparsity problem, and are limited by explicit features which are associated with objects.

C. SEQUENTIAL RECOMMENDATION

In traditional sequential recommendation, Markov chains are successful probabilistic models that model sequential patterns. The model detects sequential patterns through stochastic transitions between states. Rendle *et al.* [2] proposed a model called FPMC, combining MF and Markov chains for sequential recommendations, which showed good performance in modeling individual preferences for sequential recommendations. The model proposed by He and McAuley [3], called Fossil, extended FPMC to emphasize the importance of sequential patterns by introducing high-order Markov chains and similarity models. The result has shown a better recommendation performance than the aforementioned recommendation model.

Recently, similar to several other domains like Education and Economic, deep learning technologies have been introduced in the recommendation field, such as recurrent neural networks (RNNs) [29], [30]–[32], deep neural networks (DNNs) [33], [34], and attention mechanisms [35]–[37]. Among these techniques, RNN and CNN-based recommender systems have particularly shown a high performance for sequential recommendation. Some attention and self-attention-based recommendation systems also have shown a high performance and proven to be successful in capturing high-order sequential interaction between items and users. However, such techniques usually require a high computational time and are complex. What's more, to get high or even acceptable performance, a significant amount of data should be used for training these types of techniques.

Instead, to address the existing issues, our proposed method considers conventional machine learning-based techniques, benefiting a new content-based filtering algorithm—inspired by FISM [4]—based on the item similarity model. Additionally, we considered and extended the Fossil [3] approach to deal with sequential patterns.

III. PROBLEM DEFINITION AND NOTATIONS

Traditional CF-based recommendation models do not consider the item information and the sequential pattern to predict the list of recommendations that a user might be interested in. Generally, they focus on the user data, such as the user's profile. These methods often suffer from the data sparsity problem. We introduce the concept of the content-based filtering and sequential pattern mechanisms to alleviate these problems. The problem in this paper is constructed as follows:

Given a set of users ($\mathrm{U} = \{u_1, u_2, \ldots, u_m\}$) and a set of items ($\mathrm{I} = \{i_1, i_2, \ldots, i_n\}$), user u is related to I_u^+ and S_u^+, where $\mathrm{I}_u^+ \in I$, $\mathrm{S}_u^+ \in I$ and $\mathrm{I}_u^+ \cap \mathrm{S}_u^+ = \emptyset$, which are the item set selected by user and the set of similar items with I_u^+, respectively. The action sequence of user u is represented $S^u = (S_1^u, S_2^u, \ldots, S_t^u)$, where S_t^u denotes the t-th item selected by user u. By considering such information, the recommender system aims to predict the next item for each user, and to recommend the appropriate the list of recommendations. Table 1 shows the notations in this paper.

TABLE 1. The notations.

NOTATION	DESCRIPTION
$\mathcal{U}, \mathcal{I}$	the set of user and the set of item user
u, i, t	a user u, an item and the time step
$\mathcal{I}_u^+$	the item set selected by user u
$\mathcal{S}_u^+$	the similar item set with $\mathcal{I}_u^+$
β_i	the bias term of item i
X, Y	the latent vectors
K	the dimensionality of a low-rank matrix
L	the order of Markov chains
η	the global weight vector
η^u	the weight vector of a user u
$>_{u,t}$	the individualized total order of user u at time step t
ϵ	the learning rate
$\sigma(\cdot)$	the logistic function

IV. THE PROPOSED HYBRID RECOMMENDER MODEL

Our proposed model consists of three parts. It initially uses a content-based filtering approach based on the item similarity model to find similarity between the set of items and user's items. To do so, it uses a term frequency-inverse document frequency method (TF-IDF), which uses the textual information of the content, to possibly include more useful information about the item. It then applies an item similarity-based method to recognize user's long-term preference (to model user preference). Finally, it uses high-order Markov chain to have more meaningful interactions in the user's item sequence (to find the short-term preference of a user).

A. MODELING THE CONTENT-BASED FILTERING

As previously mentioned, we use TF-IDF method and Cosine similarity measure to find similarity between the set of items and user's items. Both TF-IDF and Cosine similarity methods are widely used methods in computer science fields for finding an item that is similar to any particular item, and then rank similar items [38]. These methods are also capable of finding groups of similar item by using clustering algorithm. Basically, the higher term frequency of a particular keyword means that a particular keyword is more important to the items than one with a lower term frequency. The IDF value is calculated by dividing the entire set of documents (items) by the amount of documents for which a particular keyword appears. If a particular keyword appears in a lot of the documents, IDF values are small, and vice versa. TF-IDF is calculated by multiplying the TF value by the IDF value, as shown in Equation (1).

$$w_{term,i} = tf_{term,i} \cdot \log\left(\frac{|\mathcal{J}|}{df_{term}}\right) \tag{1}$$

In this equation, the total amount of documents in the dataset is referred to as $|\mathcal{J}|$, the weight value for a specific term in document I is denoted as $w_{t,i}$, the number of times that a specific term appears in document I is denoted as $tf_{term,i}$, and the number of documents in the entire dataset that a particular term appears is denoted as df_{term} [39].

The item-item similarity is calculated by a similarity measure. We used the well-known Cosine similarity measure because TF-IDF outputs the vectors of items. The Cosine similarity of two item vectors is computed by the dot product of two unit vectors, as shown in Equation (2).

$$sim\left(\mathcal{J}_i, \mathcal{J}_j\right) = \frac{\mathcal{J}_i \cdot \mathcal{J}_j}{||\mathcal{J}_i|| \cdot ||\mathcal{J}_j||} \tag{2}$$

where $\mathcal{J}_i$ and $\mathcal{J}_j$ are vectors of i and j. We used the calculated similarity to find a set of items similar to that which was observed in the user profile. High similarity of items means that the contents information is similar, whereas low similarity means that the contents information is different. Our method models the relationship between the contents using a similar set of contents.

Consequently, our proposed model that is inspired by FISM [4] uses the output produced by TF-IDF method and Cosine similarity measure. The model is illustrated by Equation (3).

$$\hat{r}_{ui} \propto \langle \sum_{j \in S_u^+ \setminus \{i\}} X_j, Y_i \rangle \tag{3}$$

In this equation the item set similar to the items rated by user u is denoted as S_u^+.

B. MODELING USER PREFERENCE

In general, a recommender system models the user preference though the user profiles, such as the user-item pair. The MF is a successful method to model the user preferences with two latent-factors by factorizing the user-item matrix [40]–[42]. A user preference for the item i that presents $\hat{r}_{ui}$ is represented by the user-item latent factors (X_u and Y_i). This model is based on Equation (4), as follows:

$$\hat{r}_{ui} = \langle X_u, \; Y_i \rangle \tag{4}$$

The other method to model the user preferences is the sparse linear methods called SLIM, which presents the user preference with item-item similarity [25]. In SLIM, the user preference is represented by the summation which the ratings of the items a specific user has already rated. This method is shown in Equation (5):

$$\hat{r}_{ui} = \sum_{j \in I_u^+ \setminus \{i\}} a_{ji} \tag{5}$$

In this equation, the set of items selected by user u and the element at the j-th row and the i-th column of the item-to-item similarity matrix are denoted as I_u^+ and a_{ji}, respectively. The two items' similarities are represented by each element.

Nonetheless, SLIM weakens user-item interactions by considering item-item similarity. FISM approach introduced MF's ideas of breaking the item-to-item similarity matrix into two lower rank matrices to overcome SLIM problems. FISM is represented by Equation (6).

$$\hat{r}_{ui} = \langle \sum_{j \in j_u{}^+ \setminus \{i\}} X_i, Y_i \rangle \tag{6}$$

where X and Y are both $|I| \times K$ matrices and $K \ll |I|$. This approach produces cutting-edge performance on large sparse data sets by significantly reducing the number of parameters associated with problems in the recommender system.

C. MODELING SEQUENTIAL PATTERNS

In general, a Markov chain models sequential patterns by representing the relationship between items through the transition matrix between items, and estimating the transition probabilities through the transition matrix. For example, given the last item i, the probability of the next item j can be represented as $p(j|i)$. The transition matrix can be factorized into a pair of low-rank matrices similar to [4], [25]. An inner product between the latent vectors represents the transition probability of the item i and the item j, as shown in the following Equation (7):

$$p\,(j \,|\, i) \propto \langle M_i, N_i \rangle \tag{7}$$

In this equation, the latent vectors of the item i and the item j are denoted as M and N, respectively.

To this end, Rendle *et al.* [2] proposed a model called FPMC, combining the Markov chains and MF. He and

McAuley [3] proposed a recommendation model that emphasizes a sequential pattern, combining the high-order Markov chains and the item-item similarity models. He and McAuley [3] combined two methods into one model, using the property where the dimensions of the two methods are the same. In this approach, the transition probability when given the item sequence $(S^u_{t-1}, S^u_{t-2} \ldots, S^u_{t-L})$ and the next item j is represented by the following Equation (8).

$$p_u\left(j \mid S^u_{t-1}, S^u_{t-2}\ldots, S^u_{t-L}\right) \propto \beta_j + \langle \frac{1}{|J^+_u \ \{j\}|^{\alpha}} \sum_{j' \in J^+_u \ \{j\}} P_{j'} + \sum_{k=1}^{L} (\eta_k + \eta^u_k) \cdot P_{S^u_{t-k}}, Q_j \rangle \tag{8}$$

where $(S^u_{t-1}, S^u_{t-2} \ldots, S^u_{t-L})$ denotes the L items recently consumed by user u and $\eta^u = (\eta^u_1, \eta^u_2 \ldots, \eta^u_L)$ denotes the sequential dynamics and user preferences' weight. Similarly, the global bias is $\eta = (\eta_1, \eta_2 \ldots, \eta_L)$.

D. UNIFYING ALL MODELS

The aforementioned models in the previous sections include the user preference, sequential pattern (Markov chain) and content-based filtering (content similarity). We have combined these models into a unified model called U2CMS (Unified Collaborative and Content-based similarity models with Markov chain for Sequential recommendation). Our model is represented by Equation (9) because the three models have the same dimensions. Equation (10) denotes X in Equation (9).

$$p_u\left(j \mid S^u_{t-1}, S^u_{t-2}\ldots, S^u_{t-L}\right) \propto \beta_j + \langle X, Y_j \rangle \tag{9}$$

$$X = \frac{1}{\left|S^+_u \setminus \{j\}\right|^{\alpha_1}} \sum_{e' \in S^+_u \setminus \{j\}} X_{e'} + \frac{1}{\left|I^+_u \setminus \{j\}\right|^{\alpha_2}} \sum_{j' \in I^+_u \setminus \{j\}} X_{j'} + \sum_{k=1}^{L} \left(\eta_k + \eta^u_k\right) \cdot X_{S^u_{t-k}} \tag{10}$$

E. THE LEARNING MODEL IN THE U2CMS

To give a ranking to the observed items aimed at producing recommendation lists that might be of interest to a certain user is the main goal of sequential prediction tasks. Sequential Bayesian Personalized Ranking (S-BPR) [43] is a loss function that fits these goals. For example, to indicate that item j for user u ranks lower than item i (at step t given the action sequence before t), it uses $i >_{u,t} j$. This loss function is represented as follows:

$$p\left(S^u_t >_{u,t} j \mid \Theta\right) = \sigma\left(\hat{p}_{u,t,S^u_t} - \hat{p}_{u,t,j}\right) \tag{11}$$

where $\hat{p}_{u,t,\cdot}$ ($\cdot$ means both S^u_t and j) denotes the prediction in Equation (9) and Θ denotes the hyper parameters in the model. The hyper parameter Θ is optimized by maximum a posteriori (MAP) estimation as follows:

$$\underset{\Theta}{argmax} = ln \prod_{u \in U} \prod_{t=2}^{|S^u|} \prod_{j \neq S^u_t} p\left(S^u_t >_{u,t} j \mid \Theta\right) p(\Theta) (12)$$
$$= \sum_{u \in U} \sum_{t=2}^{|S^u|} \sum_{j \neq S^u_t} ln\left(S^u_t >_{u,t} j \mid \Theta\right) + lnp(\Theta) \tag{12}$$

In this equation, a Gaussian prior over the hyper parameters is denoted as $p(\Theta)$. A traditional training algorithm such as the stochastic gradient decent (SGD) can learn this objective function, we use this technique for estimation of the optimal hyper parameter. The learning procedure is as described below:

1. We use the dataset for uniform sampling of the user u from U, and the time step t from $\{2, 3, \ldots, |S^u|\}$.
2. The negative items $j \in I$ and $j \notin \{S^u_t, S^u_{t-1}, \ldots, S^u_{t-min(L,t-1)}\}$ are uniformly sampled.
3. The optimization procedure updates the parameters.

The time complexity of our model is O(|U|LKT), where L is the L-order Markov chains, K is the number of latent vector dimensionality, and T is the number of the iterations. However, L and K are usually small numbers for sparse datasets.

V. THE EXPERIMENTAL RESULTS

A. DATASETS

We used five amazon datasets after preprocessing (see Table 2). These Amazon.com datasets were recently collected and published by [43], [44]. These are large datasets include review texts, time stamps, and images. Table 2 shows the datasets statistics to prove the datasets diversity. We removed the data where the interaction numbers between items and users were less than five and five. After that, following previous works [3], [25], [26], [45], we changed the explicit feedback to implicit feedback. For example, where there is any interactions in profile of the user, we set rating score as 1, otherwise 0.

B. COMPARISON METHODS

To show the efficiency of U2CMS, we compared its performance with the following advanced existing approaches (Table 3 shows the properties of each model):

1. Bayesian personalized ranking matrix factorization (BPR-MF) [45] is a top-K recommendation model based on MF, which models the user preference with two latent matrices. It was considered as a default baseline method in our experiments because it does not consider sequential information and content-based filtering, and uses conventional recommendation methods.
2. The Factored Item Similarity Models (FISM) [4] is based on one of the latest recommendation algorithms to capture the relationship between items for personalized recommendations. Our proposed model was

TABLE 2. Dataset statistics.

DATASET	AUTO	BABY	GARDEN	OFFICE	VIDEO	TOTAL
#users	122,492	20,434	5,376	7,416	176,404	332,122
#items	28,473	8,293	5,098	5,490	19,421	66,775
#interaction	369,525	169,153	59,634	52,175	630,513	3,019,410
#interaction/user	3.02	8.28	11.09	9.73	3.57	3.86
#interaction/item	12.98	20.4	11.69	13.15	32.47	19

TABLE 3. Comparison table for recommendation models.

Property	[45]	[4]	[2]	[3]	[37]	[31]	U2CMS
personalized	√	√	√	√	√	√	√
sequentially aware	X	X	√	√	√	√	√
similarity-based	X	√	X	√	X	X	√
explicitly model users	√	X	√	X	X	X	X
deep neural network	X	X	X	X	√	√	X
content-based	X	X	X	X	X	X	√

inspired by this idea, combining collaborative and content-based recommendations. FISM was selected to confirm the effectiveness of sequential information and content-based filtering.

3. The Factorized Personalized Markov Chain (FPMC) [2] is a personalized item recommendation model that includes the sequential pattern with the Markov chain. However, since this method does not take into account the high-order Markov chain, it's inclusion helps showing the effectiveness of the high-order Markov chain in comparative experiments.
4. The factorized sequential prediction with item similarity models called Fossil [3] is a proposed recommendation model for sequential recommendations. It combines a user preference with the high-order Markov chains in a similarity model. This approach was inspired by FISM and FPMC, and emphasizes the role of the sequential feature to increase the performance. In our experiments, considering Fossil helps to show the effectiveness of the content-based filtering.
5. Self-Attentive Sequential Recommendation called SAS [37] introduces the adaptive self-attention mechanism method to model high-order sequence, and it shows high performance for sequential recommendations. In our experiment, SAS was considered for comparison with the self-attention mechanism-based model.
6. The recurrent translation-based network called RTN [31] is a recently proposed sparse sequential recommendation model. It unifies translation-based model and recurrent neural network. RTN was taken into account for comparison with the recent deep learning methods.
7. Our proposed model, called U2CMS, was inspired by Fossil and content-based filtering. We use a similarity model mechanism to model the high-order Markov chains and the similarity between items to emphasize the relationships between items.

On all datasets, for BPR-MF, FISM, FPMC, Fossil and U2CMS, we set the learning rate 0.02, latent dimension 100 and the number of the recommendation items 30, respectively. We adapted SAS and RTN codes from [37] and [31]. For these, we use the default setting, while customizing the evaluation metrics, and the number of the test and validation data for training.

TABLE 4. Results for evaluation measures.

Dataset	Metric	Model							% improvement
		BPR-MF	FISM	FPMC	Fossil	SAS	RTN	U2CMS	
Auto	Recall	0.039	0.083	0.026	0.082	0.086	0.100	**0.148**	48.0
	NDCG	0.017	0.048	0.014	0.04	0.031	0.041	**0.088**	83.3
Baby	Recall	0.059	0.074	0.048	0.062	0.068	0.077	**0.109**	41.6
	NDCG	0.034	0.042	0.028	0.035	0.023	0.029	**0.093**	121.4
Garden	Recall	0.045	0.086	0.044	0.082	0.126	0.126	**0.152**	20.6
	NDCG	0.022	0.047	0.025	0.046	0.044	0.050	**0.094**	88.0
Office	Recall	0.038	0.076	0.044	0.083	0.133	**0.134**	0.107	-20.1
	NDCG	0.018	0.050	0.024	0.043	0.047	0.056	**0.071**	26.8
Video	Recall	0.044	0.044	0.031	0.044	0.187	**0.193**	**0.193**	0.0
	NDCG	0.029	0.083	0.032	0.068	0.067	0.076	**0.115**	38.6
Avg.	Recall	0.045	0.073	0.039	0.071	0.120	0.126	**0.142**	12.7
	NDCG	0.024	0.054	0.025	0.046	0.042	0.050	**0.092**	70.4

C. EVALUATION METRICS

Two performance metrics were used to investigate the effectiveness of our proposed recommendation system: Recall and Normalized Discounted Cumulative Gain (NDCG). These two metrics are commonly used methodologies to measure the performance of our recommendation system [7], [19], [37]. We measured the performance with k = 30 in our experiment.

The Recall is a measure of how often users have selected the items provided by the recommender system. Equation (13) illustrates this metric:

$$Recall@k = \frac{1}{|U|}\sum_{u\in U}\frac{|G^u \cap P^u|}{|G^u|} \tag{13}$$

In this equation, for a specific user u in the test dataset, the ground-truth is G^u, while for a specific user u, the recommended list is shown by P^u.

NDCG is used as a measure of the quality of the ranking by a means of measuring the difference between ground-truth and predicted values, and has been used in several recommender system studies [6], [7], [19], [37]. DCG is measured with Equation (14) as:

$$\mathrm{DCG@k} = \frac{1}{|\mathrm{U}|}\sum_{\mathrm{u}\in\mathrm{U}}\sum_{i=1}^{k}\left(\frac{1(P_i^u \in G^u)}{\log_2(2+i)}\right) \tag{14}$$

In this equation, NDCG@k indicates that it is normalized to [0, 1]. Observe that a perfect ranking is shown by 1.

D. PERFORMANCE ANALYSIS

The aforementioned baseline methods and our method used the same dimension (D = 100) to evaluate performance via Recall and NDCG for a uniform comparison. The comparative evaluation results for the given datasets are shown in Table 4.

BPR-MF and FISM are the recommender systems that only considers user preferences. However, BPR-MF uses a method of factoring a user-item interaction matrix, and FISM factors an item-item similarity matrix. The comparison results show that FISM better reflects user preferences by highlighting the relationships between items. More specifically, FISM shows that on all datasets, the average of NDCG is 0.03 and Recall is 0.028 higher than with BPR-MF. These results show that factoring the item-item similarity matrix is technically better.

Fossil and FPMC present the user preference and the sequential patterns for the recommender system. However, Fossil highlights the sequential pattern better than FPMC, and introduces the concept of FISM. It solved a drawback in FPMC, and integrated three different models. In this regard, as illustrated in Table 4, Fossil outperforms FPMC on all the datasets; the average of NDCG is 0.021, and Recall is 0.032. That means the high-order Markov chain and item similarity method are useful for a recommender system in a sequential environment.

RTN and SAS deal with sequential information to implement a sparse sequential recommendation models using deep learning-based approaches. The comparison results between these two indicate that RTN is better than SAS on all datasets; the average of NDCG is 0.008, and recall is 0.006. These imply that both deep learning-based models are useful for sparse sequential recommendation.

The comparison between the deep learning-based recommendation methods and conventional recommendation algorithms (except U2CMS) shows that deep learning-based

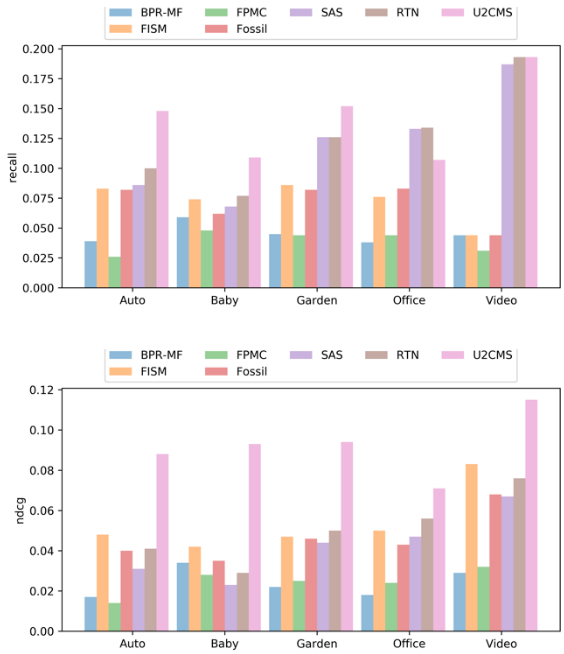


FIGURE 2. **Comparison of U2CMS to baseline methods.**

outperforms almost all conventional recommendation algorithms.

Fig. 2 and Table 4 illustrate the NDCG and Recall evaluations of the five datasets for all the methods (see comparison method section) used in the experiment. According to these results, U2CMS has an average of 0.046 for NDCG and a Recall of 0.071 which is higher than Fossil (that is considered the best performed model among traditional recommendation algorithms); an average of 0.042 for NDCG and a Recall of 0.016 that is higher than RTN (that is the best performed deep learning-based approach). Given performance of all existing methods, it can be said that U2CMS mostly has better recommendation performance than other recommendation models. This means that the content-based filtering approach employed in our proposed recommender system works well and contributes to generating more suitable recommendations.

E. THE EFFECT OF DIFFERENT N-ORDER MARKOV CHAIN

Here, we use the new term ''N-order Markov chain'' to analyze the change in performance of the high-order Markov chain ($N \in \{1, 2, 3, 4, 5\}$). In other words, we compared the performance of recommendations with different values for N. Performance comparisons were made through the 1-order Markov chain. Fig. 3 shows the performance of U2CMS with different values for N in all the datasets. An increase in the number of N elicits an increase in the recommendation performance for most datasets (Baby, Office, Garden, and Video). It means that a high-order Markov chain works well for sequential recommendation in our proposed algorithm.

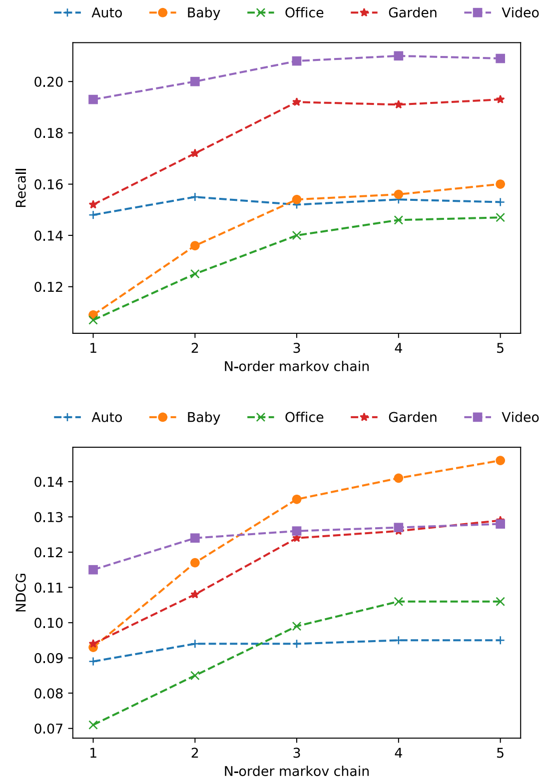


FIGURE 3. **Recommendation performance of U2CMS for different N-order Markov Chain.**

TABLE 5. **Analysis of auto, baby and garden dataset with the different number of user interaction (7 and 10).**

Dataset	#users	#items	#interactions
Auto7	1755	3941	17371
Auto10	345	1737	4775
Baby7	9267	6250	103524
Baby10	3592	4440	54807
Garden7	1509	2711	15850
Garden10	382	1128	5687

F. THE EFFECT OF SPARSITY

To show the effectiveness of our proposed approach in dealing with sparsity issue compared to the best performed traditional and deep learning-based models, we conduct another experimental work. To do so, we manipulated Auto, Baby and Garden dataset as shown in Table 5 to possibly have different sparsity with users' different minimum activity in preprocessing (i.e. 7, 10). We compare each model's performance using the same parameter value as in section V-B. The results are shown in Table 6.

Table 6 shows that U2CMS outperforms the other approaches on different sizes and the length of sequences of datasets. Especially, in small size data set and high-order sequential datasets (e.g. Auto 10 and Garden 10), U2SCMS and Fossil outperform both SAS and RTN which are deep learning-based algorithms. This shows that the conventional machine learning approach compared to deep learning-based

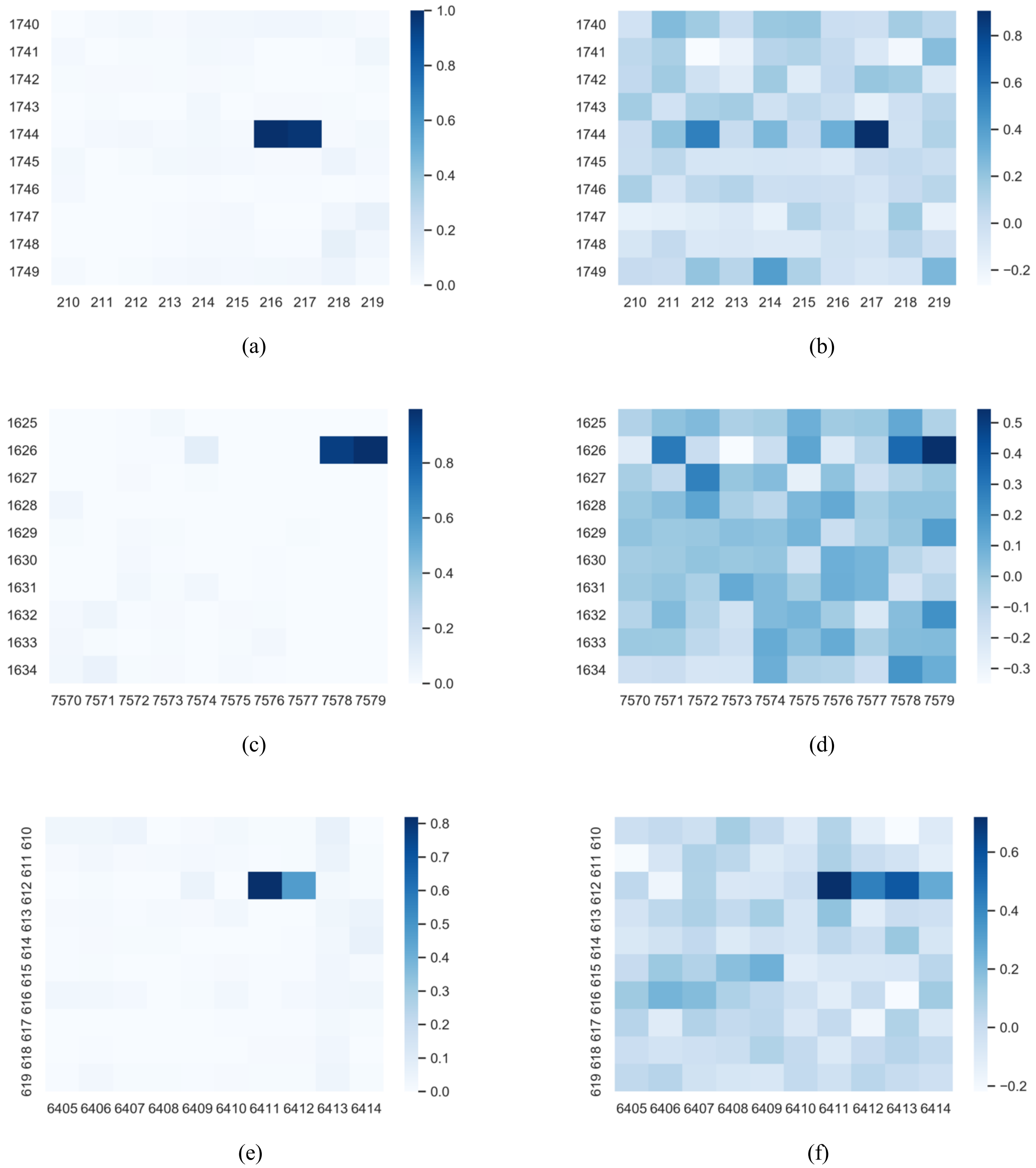


FIGURE 4. **Heat maps with cosign-similarity matrix (left) and item similarity matrix (right) on baby dataset.**

approaches appear to have a better performance on the small and high-order sequential dataset. Thus, it can be concluded that our proposed approach is mainly stable on various sparse datasets (with different size and sequence).

G. THE EFFECT OF THE CONTENT-BASED FILTERING

To show the efficiency of our approach in regard to the proposed content-based filtering model, we carried out another analysis which compares cosign-similarity matrix and item-similarity matrix on one data set (Baby dataset here).

Findings from this analysis show the accuracy of our proposed model in identifying similar items. Fig. 4 presents on the left side the cosign-similarity matrix (i.e. Fig. 4a, 4c, and 4e), whereas on the right side it shows the item-similarity matrix resulted by our proposed model (i.e. Fig. 4b, 4d, and 4f). For instance, in Fig. 4a the similarity scores

TABLE 6. Performance (NDCG@30) of four models on auto, baby and garden dataset with different the number of user interaction (7 and 10).

Dataset	Fossil	SAS	RTN	U2CMS
Auto7	0.0638	0.0379	0.0456	**0.0742**
Auto10	0.0727	0.0285	0.0482	**0.1012**
Baby7	0.0470	0.0276	0.0282	**0.0688**
Baby10	0.0566	0.0324	0.0305	**0.0689**
Garden7	0.0718	0.0560	0.0730	**0.0913**
Garden10	0.1255	0.0511	0.1282	**0.1591**

between item 1744 and 216, and item 1744 and 217 are higher than the others. Similarly, in Fig. 4b the similarity score between item 1744 and 217 is higher than the others. This indicates that our proposed model precisely identifies similar item information by using our content-based filtering model. However, similarities in item 1744 and 216 are slightly lower, meaning that these are similar items that might have a lower interaction. Another example is presented in Fig. 4c and 4d, where the similarity scores of item 1626 and 7578, and item 1626 and 7579 are similar on both figures (with almost the same intensity). This proves the success of our proposed model in correctly finding similarities between two items. Lastly, Fig. 4e and 4f, similar to the previous example, show almost the same high accuracy in identifying similar items. It is thus fair to say that our proposed model performs well in spotting similarities between items.

H. LIMITATIONS

One limitation of this work is the data type. This study did not focus on features representing contents characteristics such as utility and context-awareness. As such features have the potential to enrich the performance of the recommendation system, one could in the future work considers them while developing the model. Another limitation of this work is the similarity methods used. In this work, TF-IDF and cosine similarity methods were used, and including other types of similarity methods, such as Okapi BM25, could be an alternative way to validate the effectiveness of content-based filtering recommender systems.

VI. CONCLUSION

We proposed in this paper a hybrid recommendation system that combines collaborative and content-based similarity models with Markov chain for sequential recommendation, which we abbreviate as U2CMS. Our basic assumption is that integration of content-based filtering and sequential recommendation models can improve recommendation performance by emphasizing deep relationships between items. We decompose the content-based filtering, user preferences and sequential patterns into two sub matrices, respectively. Because all the dimensions of the matrix are the same, we combine them into one model. We use the SGD algorithm to optimize the model.

This model alleviates the sparse data and cold user problems of traditional recommender systems. In addition, qualitative and quantitative evaluations were performed using data from real commercial systems. The comparative evaluation results show that our model is superior to other baseline models. In addition, we compared the performance of recommendations with different values for N. It also showed that our model works well in sequential environments.

In our future research, we will employ deep learning techniques and state-of-the-art natural language processing techniques to expand U2CMS and improve its recommendation accuracy.

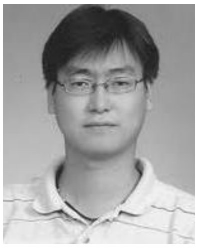
REFERENCES

[1] D. Billsus, M. J. Pazzani, and J. Chen, "A learning agent for wireless news access," in *Proc. 5th Int. Conf. Intell. user Interface*, 2000, pp. 33–36.
[2] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 811–820.
[3] R. He and J. McAuley, "Fusing similarity models with Markov chains for sparse sequential recommendation," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 191–200.
[4] S. Kabbur, X. Ning, and G. Karypis, "FISM: Factored item similarity models for top-N recommender systems," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 659–667.
[5] D. A. Cohn and T. Hofmann, "The missing link-a probabilistic model of document content and hypertext connectivity," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 430–436.
[6] Y. Yang, D. Hooshyar, J. Jo, and H. Lim, "A group preference-based item similarity model: Comparison of clustering techniques in ambient and context-aware recommender systems," *J. Ambient Intell. Hum. Comput.*, vol. 11, pp. 1441–1449, Apr. 2018.
[7] Y. Yang, D. Hooshyar, and H. S. Lim, "GPS: Factorized group preference-based similarity models for sparse sequential recommendation," *Inf. Sci.*, vol. 481, pp. 394–411, May 2019.
[8] Y. Yang, J. Jo, and H. Lim, "Unifying user preference and item knowledge-based similarity models for top-N recommendation," *Pers. Ubiquitous Comput.*, vol. 15, pp. 1–10, Jul. 2019.
[9] J. Chen, X. Zhou, and Q. Jin, "Recommendation of optimized information seeking process based on the similarity of user access behavior patterns," *Pers. Ubiquitous Comput.*, vol. 17, no. 8, pp. 1671–1681, 2013.
[10] Y. Cheng, J. Liu, and X. Yu, "Online social trust reinforced personalized recommendation," *Pers. Ubiquitous Comput.*, vol. 20, no. 3, pp. 457–467, 2016.
[11] M. Hong, J. J. Jung, F. Piccialli, and A. Chianese, "Social recommendation service for cultural heritage," *Pers. ubiquitous Comput.*, vol. 21, no. 2, pp. 191–201, 2017.
[12] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.
[13] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in *Proc. ACM Conf. Comput. Supported Cooperat. Work*, 1994, pp. 175–186.
[14] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2011, pp. 1–35.
[15] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proc. Conf. Uncertainty Artif. Intell.*, 1998, pp. 43–52.
[16] W. Pan and L. Chen, "GBPR: Group preference based Bayesian personalized ranking for one-class collaborative filtering," in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 1–5.
[17] T. V. R. Himabindu, V. Padmanabhan, and A. K. Pujari, "Conformal matrix factorization based recommender system," *Inf. Sci.*, vol. 467, pp. 685–707, Oct. 2018.

[18] V. Kumar, A. K. Pujari, S. K. Sahu, V. R. Kagita, and V. Padmanabhan, "Collaborative filtering using multiple binary maximum margin matrix factorizations," *Inf. Sci.*, vol. 380, pp. 1–11, Oct. 2017.
[19] D. Liang, J. Altosaar, L. Charlin, and D. M. Blei, "Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 59–66.
[20] M. M. Lin, B. Dong, and M. T. Chu, "Integer matrix factorization and its application," Tech. Rep., 2009.
[21] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1257–1264.
[22] F. Ortega, A. Hernando, J. Bobadilla, and J. H. Kang, "Recommending items to group of users using matrix factorization based collaborative filtering," *Inf. Sci.*, vol. 345, pp. 313–324, Jun. 2016.
[23] R. Pan and M. Scholz, "Mind the gaps: Weighting the unknown in large-scale one-class collaborative filtering," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 667–676.
[24] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 263–272.
[25] X. Ning and G. Karypis, "SLIM: Sparse linear methods for Top-N recommender systems," in *Proc. IEEE 11th Int. Conf. Data Mining*, Dec. 2011, pp. 497–506.
[26] R. J. Mooney and L. Roy, "Content-based book recommending using learning for text categorization," in *Proc. 5th ACM Conf. Digit. Lib.*, 2000, pp. 195–204.
[27] R. Van Meteren and M. Van Someren, "Using content-based filtering for recommendation," in *Proc. Mach. Learn. New Inf.*, 2000, pp. 47–56.
[28] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, 1983.
[29] B. Hidasi and A. Karatzoglou, "Recurrent neural networks with Top-k gains for session-based recommendations," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 843–852, doi: 10.1145/3269206.3271761.
[30] D. Dong, X. Zheng, R. Zhang, and Y. Wang, "Recurrent collaborative filtering for unifying general and sequential recommender," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3350–3356, doi: 10.24963/ijcai.2018/465.
[31] N. Chairatanakul, T. Murata, and X. Liu, "Recurrent translation-based network for Top-N sparse sequential recommendation," *IEEE Access*, vol. 7, pp. 131567–131576, 2019.
[32] S. Niu and R. Zhang, "Collaborative sequence prediction for sequential recommender," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 2239–2242, doi: 10.1145/3132847.3133079.
[33] X. He, X. Du, X. Wang, F. Tian, J. Tang, and T.-S. Chua, "Outer product-based neural collaborative filtering," 2018, *arXiv:1808.03912*. [Online]. Available: http://arxiv.org/abs/1808.03912
[34] J. Tang and K. Wang, "Personalized Top-N sequential recommendation via convolutional sequence embedding," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Marina Del Rey, CA, USA, 2018, pp. 565–573, doi: 10.1145/3159652.3159656.
[35] L. Mei, P. Ren, Z. Chen, L. Nie, J. Ma, and J.-Y. Nie, "An attentive interaction network for context-aware recommendations," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Turin, Italy, Oct. 2018, pp. 157–166, doi: 10.1145/3269206.3271813.
[36] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Singapore, 2017, pp. 1419–1428, doi: 10.1145/3132847.3132926.
[37] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 197–206, doi: 10.1109/ICDM.2018.00035.
[38] W. Zhang, T. Yoshida, and X. Tang, "A comparative study of TF*IDF, LSI and multi-words for text classification," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2758–2765, Mar. 2011.
[39] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.
[40] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
[41] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, pp. 1–19, Oct. 2009.
[42] Y. Cheng, L. Yin, and Y. Yu, "LorSLIM: Low rank sparse linear methods for Top-N recommendations," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2014, pp. 90–99.
[43] T. Zhao, J. McAuley, and I. King, "Leveraging social connections to improve personalized ranking for collaborative filtering," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, 2014, pp. 261–270.
[44] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2015, pp. 43–52.
[45] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. Conf. Uncertainty Artif. Intell.*, 2011, pp. 452–461.

YEONGWOOK YANG received the master's degree in computer science education and the Ph.D. degree from the Department of Computer Science and Engineering, Korea University, Seoul, South Korea. He is currently a Senior Researcher with the University of Tartu, Tartu, Estonia. His research interests include information filtering, recommendation systems, educational data mining, and deep learning.

HONG-JUN JANG received the B.Sc. and Ph.D. degrees in computer science education from Korea University, Seoul, South Korea. His research interests include data mining, machine learning, and data base.



BYOUNGWOOK KIM received the B.Sc. and Ph.D. degrees in computer science education from Korea University, Seoul, South Korea. Since 2018, he has been with the Department of Computer Engineering, Dongguk University, Gyeongju, South Korea. He had been with the Creative Informatics and Computing Institute, Korea University, Seoul. He is currently a Research Fellow with the New Media Education and Development Center, Korea National Open University. His research interests include data mining, machine learning, intelligent tutoring systems, and e-learning systems.