Information Sciences 481 (2019) 394–411

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

GPS: Factorized group preference-based similarity models for sparse sequential recommendation

Yeongwook Yang[a], Danial Hooshyar[b], Heui Seok Lim[a,*]

[a] Department of Computer Science and Engineering, Korea University, Seoul, the Republic of Korea
[b] Institute of Education, University of Tartu, Estonia

ARTICLE INFO

Article history:
Received 13 February 2018
Revised 21 December 2018
Accepted 22 December 2018
Available online 3 January 2019

Keywords:
Recommender systems
Sequential recommendation
Similarity models
Group preference

ABSTRACT

One of the key tasks for recommender systems is the prediction of personalized sequential behavior. There are two primary means of modeling sequential patterns and long-term user preferences: Markov chains and matrix factorization, respectively. Together, they provide a unified approach to predicting user actions. In spite of their strengths in tackling dense data, however, these methods struggle with the sparsity issues often present in real-world datasets. In approaching this problem, we propose combining similarity-based methods (demonstrably helpful for sequentially unaware item recommendation) with Markov chains to offer individualized sequential recommendations. This approach, called GPS (a factorized group preference-based similarity model), further leverages the idea of group preference along with user preference to introduce a greater array of interactions between users—which in turn eases the problem of data sparsity and cold users and cuts down on the assumption of a strong independency within various factors. By applying our method to a range of large, real-world datasets, we demonstrate quantitatively that GPS outperforms several state-of-the-art methods, particularly in cases with sparse datasets. Regarding qualitative findings, GPS also grasps personalized interactions and can provide recommendations that are both on-target and meaningful.

1. Introduction

The rise and expansion of e-commerce has greatly expedited the convenience of online transactions. Buyers now have nearly unlimited access to products and related information, and this access has served to significantly change traditional conceptions of product purchasing. As the variety of available products increases, however, so does the need for a means by which to identify and understand which product most closely fits an individual's tastes. For this reason, activities such as modeling and grasping interactions between users and products (and also the relations between products) are vitally important [3,12]. What is needed is a system that offers both item-to-user recommendations ("What product would please this particular user?") and item-to-item recommendations ("Which shoes would pair well with the pants that this user just bought?"). These interactions represent two different temporalities: long-term user preferences on the one hand, and immediate sequential patterns on the other.

Both long- and short-term factors must be considered in regard to forecasting individualized sequential behavior from collaborative data, such as a users' shopping history, as the prediction must account for both individualized and sequential

* Corresponding author
E-mail address: limhseok@korea.ac.kr (H.S. Lim).

https://doi.org/10.1016/j.ins.2018.12.053
0020-0255/© 2019 Elsevier Inc. All rights reserved.

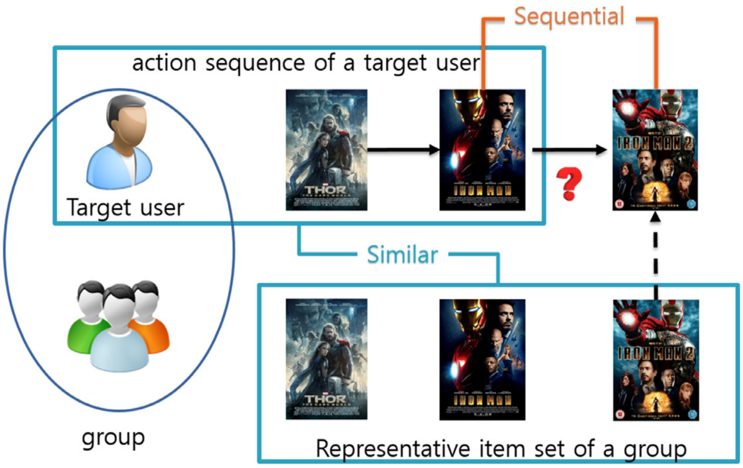

Fig. 1. Example of GPS recommendation.

aspects [34]. This difficulty is compounded by the sparsity present in many real-world datasets. In particular, limited training sequences allow for little confidence in estimating parameters [5,8]. However, this problem is ignored by models that only consider historical temporal dynamics while neglecting the sequential patterns of specific users. (In other words, such models would only consider the popularity of *Twilight* between 2010 and 2014, without examining what *Twilight* viewer Lisa will watch next after finishing the movie).

To date, modelers have followed two paths in charting user preferences: traditional item recommendation algorithms and item similarity-based algorithms. The first of these relies on matrix factorization in which any given user or item is assigned a numerical vector of the same dimension, so that their compatibility is gauged by the inner product of their representations [27]. The second, on the other hand, provides a particular user with recommendations according to the similarity of new items to those he or she has previously bought or liked. Although item-to-item similarity-based algorithms such as factored item similarity models (FISM) do not develop specific parameters for each user, they substantially outperform competing baselines (matrix factorization included), particularly when datasets are sparse [13,18].

None of the aforementioned models take sequential dynamics into account, however. For this purpose, we must employ other approaches capable of capturing sequential patterns—Markov chains, for example. With this in mind, Rendle, Freudenthaler, and Schmidt-Thieme [40] put forward factorized personalized Markov chains (FPMC). By modeling user preferences as well as sequential behavior, such models outperform sequential and general recommenders alike. The positive results of FPMC notwithstanding, the fact that it combines its components linearly suggests that it makes strong independent assumptions between various factors, such as the idea that every component independently affects a user's subsequent purchase. He and McAuley [13] have put forth a new model called FOSSIL which combines the strong points of the previously mentioned approaches. Additionally, it adds Markov chains and considers the similarity between items in order to address sequential dynamics and the sparsely populated datasets encountered in the real world. Although FOSSIL surpasses the other two methods, it is limited in considering only the preference of the item that the user chooses during the learning period. It thus has limited performance in offering recommendations in the absence of user-selected items.

We develop in this paper a FOSSIL-inspired hybrid approach, which conjoins similarity-based techniques with the Markov chain to address real-world datasets in which data is sparse and dynamics are sequential. In contrast to all of the previous approaches, our model takes both the sequential dynamics and group preference into account to deepen user-interaction. It uses higher-order Markov chains to model smooth sequentiality over a number of time steps, and models group preference by calculating the similarity between users, clustering similar users, and extracting representative item sets of groups. What results is GPS: a factorized group preference-based similarity model for sparse sequential recommendations. GPS *integrates three different methods—user and group preferences and sequential activity—by acquiring an individualized weighting scheme over a sequence of items that classifies both users and groups by preference and strength of sequential behavior.* GPS thus promises both to address the problem of the cold user by placing any user within the parameters derived from historical items (provided the representations of items can be accurately estimated) and to more heavily weigh short-term dynamics in cases where "global" sequential patterns are readily available. In doing so, GPS substantially addresses the issue of sparsity since it can make reasonable predictions even in the absence of observed actions for a specific user. Furthermore, GPS can capture a greater depth of interactions among users because it joins group preference to models of item similarity. This reduces the effects of data sparsity and the acuteness of the cold-start problem, and thus enhances the performance of the algorithm (Fig. 1 demonstrates an example of a GPS recommendation).

GPS thus makes the following contributions: 1) it offers a new technique that joins group preference to item similarity-based models; 2) it promises, by combining user and group preference, to alleviate problems in data sparsity and cold users, and reduces assumptions of strong independence among various factors; 3) on a range of sizeable, real-world datasets, it demonstrably surpasses a range of cutting-edge algorithms; and 4) it can visualize a learned model and thus analyze the dynamics captured (both sequential and individualized).

2. Literature review

Here, we discuss a number of pertinent models: those that concern methods of item recommendation that model user preferences but fail to account for sequential dynamics; those that address the matter of sequential prediction but focus only on item similarity (at the expense of information from other users) or that assume strong independence among various factors; those that address temporal dynamics but are dependent on clear time stamps; and those that apply group and user preferences in item recommendation for the purpose of gaining greater depth of interaction among users.

2.1. Item recommendation

Typically, item recommendation depends on collaborative filtering (CF) to learn from direct feedback mechanisms such as star-ratings [10]. Collaborative filtering can be classified into two primary categories: memory-based and model-based approaches [1,45]. The former furnishes recommendations by locating k-nearest-neighbors, according to given similarity measures, for specific users or products [28]. The latter derives recommendations by factorizing the user-item correlation matrix. Lee, Jun, Lee, and Kim [25], for instance, handle the market basket data as a binary matrix of items and users, and thus arrive at recommendations by treating it with a binary logistic regression model that uses principal component analysis. In contrast, Hu, Koren, and Volinsky [15] use least-square optimization to factorize user-item pairs, and then control the significance of observations through pair confidence. Likewise, Pan and Scholz [35] add weights to user-item pairs, seeking optimal factorization by employing criteria of least square loss and hinge loss. *There are a number of these types of algorithms: for example, Bayesian methods* [4,38]*, restricted Boltzmann machines* [44]*, and the matrix factorization (MF) methods upon which a number of top-of-the-line recommendation approaches are based* [14,24,31]. These approaches adequately capture the overall tastes of users, but they have no means of catering recommendations to the recent purchases of particular users if they do not also model sequential behavior [46].

2.2. Sequential recommendation

Markov chains offer us an excellent way to determine the stochastic transitions between 'states' . A number of earlier works examine the strength of Markov chains within sequential recommendation domains for discovering sequential patterns (e.g., [49,19]) and modeling decision processes [43]. Such models fail to account for the information of other users, however, and focus only on the similarity between items. Rendle et al. [40] have more recently suggested conjoining Markov chains' great capacity for smoothly modeling sequential actions with matrix factorization, which excels in modeling individual preferences for sequential recommendation. This combination—abbreviated as FPMC—demonstrates increased predictive power because of these combined strengths. A serious problem persists in FPMC, however, in that it makes strong independent assumptions among a number of different factors, as if each component of the model exerts a distinct effect on the users' next purchase. While our work runs in this same vein, it adds a pair of important elements: first, we employ a similarity-based method for both group and user preference modeling, to diminish sparsity issues and to avoid such strong independent assumptions among the various factors; second, we also include Markov chains with higher orders to model sequential smoothness over many time steps.

2.3. Modeling temporal dynamics

Outside of the visual domain, there have been various attempts within the field of machine learning to study temporally evolving data through algorithms such as support vector machines [6], decision trees [17], and instance-based learning [2]. Likewise, collaborative filtering approaches akin to our own grapple with temporal dynamics. Specifically, there have been some works using similarity-oriented collaborative filtering which arrive at similarities by employing a time-weighting approach that applies decaying weights to previously rated items (e.g., [11]). More recent works depart from this approach, however, in favor of the matrix factorization method [22]—such as Koren's [21] promising results in applying matrix factorization to Netflix data in order to model the embedded temporal dynamics. That said, this approach relies directly on specific time stamps as it constructs models to make past actions intelligible. This is a different task than the one required of sequential prediction, since the latter models not time-stamped actions but sequential relationships.

2.4. Group preference

There is still a great need for developing a recommendation system that takes sequential information into account: such a system will promote a decision support system that is able to utilize sequential information [36]. Here, we put forward an approach that employs clustering techniques to produce recommendations that take into account both sequential information and content. We first ascertain user similarities by way of a hybrid similarity measure (or S^3M) that takes into account both sequence and content. We subsequently employ these similarity scores to cluster together groups with similar user profiles and to derive representative items from these groups. Consequently, we are able to capture deeper layers of interactions among users, and thereby to minimize the dilemma of data sparsity, the cold-user problem, and any related performance difficulties in the algorithm.

3. The proposed model

3.1. Problem formulation and notation

Most recommendation methods are wholly focused on modeling the types of objects that might interest each user but do so without taking into consideration any pertinent sequential information—such as the most recent item the user purchased or reviewed, the places the user has recently visited, and so forth. Moreover, these systems of recommendation face the dilemma of data sparsity, since nonrated items vastly outnumber rated items for users. The way we address these issues is to apply group preference-based similarity models to sequential prediction. The problems are configured as follows:

Given a set of users and items, we can represent a set of users denoted as $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$ and a set of items denoted as $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$. Each user u is associated with a sequence of actions $S^u = (S_1^u, S_2^u, \ldots, S_t^u)$, where S_t^u denotes the t-th selected item of user u. We use $\mathcal{I}_u^+$ to denote the set of items in S_t^u, where the sequential information is ignored. All users u are clustered with similar users, and a set of clusters is denoted $\mathcal{G} = \{g_1, g_2, \ldots, g_c\}$. g_c conveys the representative items of a cluster. $\mathcal{G}_g^+$ denotes the representative items in g_c, where $\mathcal{G}_g^+ \in \mathcal{I}$.

Using these symbols, we seek in this study to forecast the next item each user will choose and to provide the user with suitable recommendations (see Table 1 for the notations used in this paper).

3.2. Modeling the user preferences

Essentially, the recommendation system models the user preferences based on the information concerning the user-chosen items. The MF has been a favored means by which to accomplish this within the traditional recommendation systems. The MF begins with a low-rank assumption [42], positing the users and items within a k-dimensional low-rank latent space. The MF then approximates the user preference for the given items as the inner products of the user and item latent factors. In the cases where the preference of the user u for the item i is $\hat{r}_{ui}$, the user preference is modeled by setting the user latent factor to P_u and the item latent factor to Q_i according to Eq. (1), as follows:

$$\hat{r}_{ui} = \langle P_u, Q_i \rangle \tag{1}$$

At present, one of the other approaches used in this research is called SLIM (sparse linear methods), which gauges user preference for a given item by way of item-to-item similarity [32]. User preference is represented within SLIM by $\hat{r}_{ui}$, arrived at by aggregating the ratings of the items user u has rated. That is, SLIM arrives at the preferences of user u for item i by way of an item-to-item similarity matrix A, which it learns and develops according to the user history as shown in Eq. (2):

$$\hat{r}_{ui} = \sum\nolimits_{j\in\mathcal{I}_u^+\setminus\{i\}} a_{ji} \tag{2}$$

where $\mathcal{I}_{\boldsymbol{u}}^+$ is the set of items purchased by u. $\mathbf{a_{ji}}$ is the element at the j-th row and the i-th column of the item-to-item similarity matrix A. It indicates the similarity between the two items, j and i. However, while SLIM has demonstrated strong results, its initial model examines a great quantity of item-to-item matrices, leading to minimal user-item interactions. This problem has been resolved in the latent version of SLIM [7] by way of discovering the sparsity property of A through using L1 norm regularization in order to establish the parameters.

FISM (factored item similarity model) departs from SLIM by focusing on a way to approximate the similarity matrix by decomposing A to a result of two low-rank matrices for the problem. FISM arrives at A as shown in Eq. (3):

$$\hat{r}_{ui} = \left\langle \sum\nolimits_{j\in\mathcal{I}_u^+ \setminus \{i\}} P_j, Q_i \right\rangle \tag{3}$$

where P and Q are both $|\mathcal{I}| \times K$ matrices and $K \ll |\mathcal{I} \ll$. The advantage of this approach is that it greatly reduces the number of parameters used and generates state-of-the-art performance on a number of sparse datasets. Moreover, in contradistinction to SLIM, it better depicts the transitive relations between items. These improvements add up to a strong overall recommendation performance, even within sparse real-life datasets.

3.3. Modeling group preferences

Individual preference assigns a score to a particular user's preference for an item, while *group preference* charts the cumulative score of a group of users' preferences for an item. This latter preference is an essential component in forecasting user preference concerning a particular item, as it injects richer interactions among users and therefore decreases the dependency on *individual* and *independence* assumptions [37]. To date, however, group preference did not take into account the sequential information of users. We put forth here a new group preference model, one which accounts for the sequential behavior of users. It is comprised of four steps:

Table 1
Notations.

Notation	Explanation
$\mathcal{U}, \mathcal{I}, \mathcal{G}$	User set, item set, group set
$\boldsymbol{u}, \boldsymbol{i}, \boldsymbol{g}, \quad \boldsymbol{t}$	A specific user, item, group, time step
$\boldsymbol{S_t^u}$	The item user u interacted with at time step t
$\boldsymbol{S^u}$	Action sequence of user u
$\mathcal{I}_{\boldsymbol{u}}^{+}$	the set of items in S^u
$\mathcal{G}_{\boldsymbol{g}}^{+}$	the set of representative items in group g
$\boldsymbol{\beta_i}$	Bias term associated with item
$\boldsymbol{P_i}$	Latent vector associated with item i
$\boldsymbol{Q_i}$	Latent vector associated with item i
$\mathbf{K}$	Dimensionality of the vector representing each user/item
$\mathbf{L}$	Order of Markov chains
$\boldsymbol{\eta}$	Global weighting vector
$\boldsymbol{\eta}^{\mathbf{u}}$	Personalized weight vector
$\mathbf{p}_{\mathbf{u}(\mathbf{j}\vert\mathbf{i})}$	Probability that user u chooses item j after item i
$\hat{\mathbf{p}}_{\boldsymbol{u},\boldsymbol{t},\boldsymbol{i}}$	Prediction that user u chooses item I at time step t
$>_{\mathbf{u,\ t}}$	Personalized total order of user u at time step t
$\boldsymbol{\alpha}$	Weighting factor
ϵ	Learning rate
$\boldsymbol{\sigma}(\cdot)$	the logistic function

Table 2
Algorithm 1 Longest Common Subsequence

```
Input: sequences S^1 = (i_1, i_2, ..., i_n), S^2 = (j_1, j_2, ..., j_m),
Output: the length of an LCS L(n,m)
  an m × n Matrix L ← 0
for k from 1 to n do
    for l from 1 to m do
        if i_k = j_l then
            L(k, l) ← (L(k − 1, l − 1) + 1
        else
            L(k, l) ← max(L(k, l − 1), L(k − 1, l))
        end if
    end for
end for
return L(n, m)
```

1. Calculating user similarity.
2. Clustering similar users.
3. Extracting a representative item set of groups.
4. Developing a model for group preference.

3.3.1. Calculating user similarity

Within a recommender system field, indices of user-based similarity are typically employed in order to cluster users together according to the items they have chosen [47]. This is true both of content-based and similarity-based measures. Whereas the former seek to ascertain the similarity of content among users, the latter seek to ascertain their sequence similarity. Similarity measures such as Jaccard and Dice typify content-based approaches; sequence-based ones include measures such as the Levensthein distance, Hamming distance, and longest common sub sequence (LCS). By conjoining these two types of similarity measures, one can arrive at a hybrid measure capable of capturing the similarity among users both in terms of content and in terms of sequence. For this reason, we employ a hybrid measure (S^3M) that can account for both types of similarity. S^3M combines the Jaccard similarity measure with a sequence similarity measure based on the length of longest common subsequence (LLCS) in a linear manner. In the present study, we cluster users by means of the results of S^3M.

Another measure, the SeqSim (sequence similarity measure), employs an LCS algorithm to ascertain the similarity between two item sequences by uncovering the longest common subsequence. Let two sequences of users be defined as follows: $S^1 = (i_1, i_2 ..., i_n)$ and $S^2 = (j_1, j_2 ..., j_m)$, where i and j are the items. Given two sequences, LLCS is calculated with Algorithm 1 (see Table 2).

In addition, SeqSim is formulated as shown in Eq. (4):

$$\mathrm{SeqSim}\left(S^1,\ S^2\right) = \frac{\mathrm{LLCS}}{\max\left(|S^1|, |S^2|\right)} \tag{4}$$

where LLCS is the length of LCS.

The content-based similarity measure (ConSim) is based on the Jaccard similarity measure as shown in Eq. (5):

$$\mathrm{ConSim}\left(S^1,\ S^2\right) = \frac{\left|S^1 \cap\ S^2\right|}{|S^1 \cup\ S^2|} \tag{5}$$

Therefore, we have determined S^3M when the two sequences of items are taken as shown in Eq. (6):

$$S^3M\ =\ p\ \times\ \mathrm{SeqSim}\left(S^1,\ S^2\right) +\ (1-p) \times\ \mathrm{ConSim}\left(S^1,\ S^2\right) \tag{6}$$

where p is the weight parameter. As seen, S^3M is easily implemented and is not complex.

Algorithm 1 takes time *O(mn)*, where m is the size of sequences S^2, and n is the size of sequence S^1.

3.3.2. Clustering similar users

By computing the S^3M similarity (Section 3.3.1), we arrive at a user-by-user similarity matrix that can then be employed as an input for a clustering algorithm that groups users together. Algorithm 2 (see Table 3) demonstrates the K-means clustering algorithm that we employ: it is a highly regarded method for clustering similar users into groups [20].

Algorithm 2 takes time $O(nkl)$, where n is the size of datasets (the number of users), k is the number of clusters, and l is the number of iterations, because the centroids are recalculated many times before the algorithm converges.

3.3.3. Extracting representative item sets of groups

Having sorted similar users into groups, we extract representative item sets for these respective clusters. To accomplish this, we ascertained the frequency with which items appeared in a cluster and arranged them according to this frequency. The top N of such a list establishes the representative item set for a group (and is reiterated for every other group).

Table 3
Algorithm 2.

Algorithm 2 K-means Clustering algorithm
Input: the number of clusters k, the datasets $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x_n}\}$
Output: a set of k clusters
1: Initialize cluster centroids $M = \{\mu_1, \mu_2, \ \ldots, \ \boldsymbol{\mu_k}\} \in R^{\boldsymbol{n}}$
2: repeat
for i from 1 to n do
3: $\mathbf{c}^{\boldsymbol{i}} := \boldsymbol{d}(\boldsymbol{x_i}, \ \boldsymbol{\mu_j})$
4: for j from 1 to k do
5: $\mu_{\mathbf{j}} := \frac{1}{\lvert \boldsymbol{\mu_j} \rvert} \sum_{\boldsymbol{x_i} \in \boldsymbol{\mu_j}} \boldsymbol{x_i}$
6: until M convergence
7: return k clusters

*$d(x_i, \mu_j) = \underset{j}{\operatorname{argmin}} \lVert x_i - \mu_j{}^2 \rVert$ means the Euclidean distance method between x and μ.

For example, let g be a group and $\{i_1, \ i_2..., \ i_{10}\}$ be the items in c. $i_1 = 10, \ i_2 = 2, \ i_3 = 4, \ i_4 = 15, \ i_5 = 12, \ i_6 = 9, \ i_7 = 6, \ i_8 = 8, \ i_9 = 14,$ and $i_{10} = 11$ are the frequencies of the items. The result of sorting is $\{i_4, \ i_9, \ i_5, \ i_{10}, \ i_1, \ i_6, \ i_8, \ i_7, \ i_3, \ i_2\}$. If N is 3, we obtain a representative item set $\mathcal{G}_g^+ = \{i_4, i_9, i_5\}$. We extract the representative item sets for all groups.

3.3.4. Developing a model for the group preference

Because modeling the group-preference items entails that a representative item set is extracted for each group, it becomes possible to impart deeper interactions among the users, thereby diminishing the data-sparsity and cold-user problems. In this respect, the proposed approach builds on the fundamental concept underpinning the FISM. The group-item interaction matrix provides the group preference that is generated according to Eq. (7), as follows:

$$\hat{r}_{gi} = \left\langle \sum\nolimits_{j \in \mathcal{G}_g^+ \setminus \{\ i\}} X_j, Y_i \right\rangle \tag{7}$$

3.4. Modeling the sequential patterns

When *i* is given as the last item, the Markov chain, which is typically used to model the sequential patterns, produces a probable next item, *j*, as $p(j|i)$ according to the maximum likelihood estimation (MLE) of the item-to-item transition matrix. Akin to the FISM, the Markov chain can factorize the transition matrix into a pair of low-rank matrices; thus, the probability of the transition between the item *i* and the item *j* can be gauged using the inner product, as seen in the following Ep. (8):

$$p(j|i) \propto \ \langle \ M_i, \ N_j \ \rangle, \tag{8}$$

where M and N define the latent vectors of the *i* and the *j*. Using this formulation, it is possible to predict the interitem sequential pattern. Rendel et al. [40] recently advanced the *factorized, personalized Markov-chain model* (FPMC), where the first-order Markov chains and matrix factorizations are conjoined for the handling of the sequential information in the recommendation-system field. This approach employs the following formula, represented by Ep. (9), to predict the probability of the transition from the user-selected last-item *j* to the next-item *i*:

$$p(j|i) \propto \ \langle \ X_u, \ Y_j \ \rangle \ + \ \langle \ M_i, \ N_j \ \rangle \tag{9}$$

The first inner product calculates a user's preference for item *j*, while the second inner product calculates item *j*'s similarity to item *i*. However, because all elements of FPMC are linearly combined, this model falls into the trap of making strong independent assumptions among the various factors (such that each component exerts a discrete influence on the user's next purchase).

On the other hand, He and McAuley [13] propose an alternative method of sequential prediction to FPMC called FOSSIL, in which a similarity-based approach is modified by Markov chains. They amplify the basic idea of FISM and FPMC to address the problems of sparsity and long-tailed distributions in real-world datasets. They initially formulate their approach as shown in Eq. (10):

$$p_u(j|i) \propto \sum_{j' \in \mathcal{I}_u^+ \setminus \{\ j\}} \left\langle \ P_{j'}, Q_j \right\rangle + (\eta + \eta_u) \cdot \langle M_i, N_j \rangle \tag{10}$$

where each user is parameterized to η_u in order to control the weight of user preferences and sequential dynamics. η is a global parameter shared by η_u.

This equation connects four vectors to the item in question. These vectors can be reduced to $P = M$ and $Q = N$ since their dimensions are identical. Moreover, Ruining et al. add the term β_j to account for bias, and model users by way of high-order

Markov chains. The resulting formula follows in Eq. (11):

$$p_u\left(j|S^u_{t-1}, S^u_{t-2}..., S^u_{t-L}\right) \propto \beta\_j + \left\langle \frac{1}{|\mathcal{I}_u^+ \setminus \{j\}|^{\alpha}} \sum_{j' \in \mathcal{I}_u^+ \setminus \{j\}} P_{j'} + \sum_{k=1}^{L} \left(\eta_k + \eta^u_k \right) \cdot P_{S^u_{t-k}}, Q_j \right\rangle \tag{11}$$

where $(S^u_{t-1}, S^u_{t-2}..., S^u_{t-L})$ calculates the L items that the user u has recently consumed, and each user is integrated into $\eta^u = (\eta^u_1, \eta^u_2..., \eta^u_L)$. Similarly, the global bias is $\eta = (\eta_1, \eta_2..., \eta_L)$. FOSSIL demonstrates top-of-the-line performance in recommendations but suffers in the absence of user-selected items because it only considers the item preference with respect to the user's selections during learning and therefore offers little when the user selects few or zero items.

3.5. GPS model

The discussion of the previous sections (3.1–3.4) comprises the following three models: user preference, group preference, and sequential patterning (or Markov chain). The merging of these models produced the "GPS" hybrid model of this study that predicts the individualized preference of certain users for specific items. Because these models contain two vectors (see Eq. (3), (7), and (8)) that assume the direction of $|I| \times K$, a second item vector can be presented in each model, thereby producing the new formulation of Eq. (12), as follows:

$$\begin{aligned} &p_u\left(j|S^u_{t-1}, S^u_{t-2}..., S^u_{t-L}\right) \propto \beta_j + \left\langle P, Q_j \right\rangle \\ &P = \frac{1}{|\mathcal{G}_g^+ \setminus \{j\}|^{\alpha_1}} \sum_{e' \in \mathcal{G}_g^+ \setminus \{j\}} P_{e'} + \frac{1}{|\mathcal{I}_u^+ \setminus \{j\}|^{\alpha_2}} \sum_{j' \in \mathcal{I}_u^+ \setminus \{j\}} P_{j'} + \sum_{k=1}^{L} \left(\eta_k + \eta^u_k\right) \cdot P_{S^u_{t-k}} \end{aligned} \tag{12}$$

3.5.1. The learning model

We primarily aim to assign as high a ranking as possible to observed items, in order to help the recommender system deliver strong recommendations. Doing so produces a personalized ranking $>_{u,t}$(at each step t) to keep ranking losses at a minimum (S-BPR' [48]). By assuming independence among users and time steps, we deduce model parameters Θ by optimizing the following maximum a posteriori (MAP) estimation in Eq. (13):

$$\begin{aligned} \underset{\Theta}{\operatorname{argmax}} &= \ln \prod_{u \in U} \prod_{t=2}^{|S^u|} \prod_{j \neq S^u_t} p(S^u_t >_{u,g,t} j|\Theta) p(\Theta) \\ &= \sum_{u \in U} \sum_{t=2}^{|S^u|} \sum_{j \neq S^u_t} \ln \left(S^u_t >_{u,g,t} j|\Theta\right) + \ln p(\Theta) \end{aligned} \tag{13}$$

where g is the group of the user u, t is the time step that allows t to run from 2 to the last item in the S^u, and $p(\Theta)$ is the Gaussian prior over the model parameters.

The probability that the user, u, prefers the higher ranking of the item S^t_u compared with the "negative" item j, given the u's group, g, and the time step, t, is given by Eq. (14), as follows:

$$p(S^u_t >_{u,g,t} j|\Theta) = \sigma\left(\hat{p}_{u,g,t,S^u_t} - \hat{p}_{u,g,t,j}\right), \tag{14}$$

where $\sigma(\cdot)$ is the sigmoid function.

The widely used *stochastic gradient decent* (SGD) algorithm was followed here to optimize the objective function in the Eq. (13). The SGD training procedure works as follows: The dataset is used for the uniform sampling of the user, u, and the time step, t, from $\{2, 3, \ldots, |S^u|\}$, and then for the negative items $j \in I$ and $j \notin \{S^u_t, S^u_{t-1}, .. ., S^u_{t-\min(L,t-1)}\}$. Since the group, g, and the representative item set, $\mathcal{G}_g^+$, are determined by the user, u, it is possible to apply the optimization procedure according to Eq. (15), as follows:

$$\Theta \leftarrow \Theta + \varepsilon \left(\sigma\left(\hat{p}_{u,g,t,j} - \hat{p}_{u,g,t,S^u_t}\right) \frac{\partial \left(\hat{p}_{u,g,t,S^u_t} - \hat{p}_{u,g,t,j}\right)}{\partial \Theta} - \lambda_\Theta \Theta \right), \tag{15}$$

where ε is the learning rate, and λ_Θ is the regularization hyperparameter.

Algorithm 3 (see Table 4) represents the learning process.

The complete steps of learning the model parameters are depicted in Table 4. The time complexity of the update rule of Eq. (15) is O $(LK|\mathcal{G}|)$, where L are L-order Markov chains, K is the number of latent dimensions, and $|\mathcal{G}|$ is the size of the user group. The total time complexity is O $(T|\mathcal{U}| LK|\mathcal{G}|)$, where T is the number of iterations, and $|\mathcal{U}|$ is the number of users. However, L, K, and $|\mathcal{G}|$ are usually small numbers for sparse datasets; thereby, the computational cost is manageable.

4. Experimental evaluation

4.1. Dataset

To discern the extent to which the proposed model is capable of tackling various scenarios, an array of sizeable datasets was encompassed to enable the prediction of a range of actions (e.g., next movie to view, next item to buy, etc.). Likewise,

Table 4
Algorithm 3.

Algorithm 3 Adaptive Bayesian personalized ranking algorithm

Input: Training Data T of observed feedback
Output: The learned model parameters $\boldsymbol{\Theta} = \{\mathbf{P}_{\mathbf{i}\in\mathcal{I}},\ \mathbf{Q}_{\mathbf{j}\in\mathcal{I}}, \beta_{\mathbf{j}\in\mathcal{I}}\}$
1: initialize $\boldsymbol{\Theta}$
2: repeat
3: draw (u, t, $\mathbf{S_t^u}$) uniformly from T
4: draw j uniformly from $\mathcal{I} \setminus \mathcal{I}_{\mathbf{u}}^{+}$
5: $\mathbf{g} \leftarrow \mathbf{g_u} \in \mathcal{G}$
6: $\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} + \varepsilon\left(\sigma\left(\hat{\mathbf{p}}_{\mathbf{u},\mathbf{g},\mathbf{t},\mathbf{j}} - \hat{\mathbf{p}}_{\mathbf{u},\mathbf{g},\mathbf{t},\mathbf{S_t^u}}\right) \frac{\partial(\hat{\mathbf{p}}_{\mathbf{u},\mathbf{g},\mathbf{t},\mathbf{j}} - \hat{\mathbf{p}}_{\mathbf{u},\mathbf{g},\mathbf{t},\mathbf{S_t^u}})}{\partial\boldsymbol{\Theta}} - \lambda_{\boldsymbol{\Theta}}\ \boldsymbol{\Theta}\right)$
7: until convergence
8: return $\boldsymbol{\Theta}$

Table 5
Datasets.

Dataset	#users	#items	#actions	#actions/user	#actions/item
Amazon-Auto	122,492	28,473	369,525	3.02	12.98
Amazon-Video	176,404	19,421	630,513	3.57	32.47
Amazon-Elec	197,671	70,621	1738,410	8.79	24.62
Amazon-Office	7416	5490	52,175	9.73	13.15
Epinions	10,258	3807	30,577	2.98	8.03
Foursquare	34,686	5808	261,132	7.53	44.96
Total	876,523	243,749	4464,076	N/A	N/A

Table 6
Models.

Property	BPR_MF	FISM	FPMC	FOSSIL	GPS
personalized	√	√	√	√	√
sequentially aware	X	X	√	√	√
similarity-based	X	√	X	√	√
explicitly model users	√	X	√	X	X
high-order Markov Chains	X	X	X	√	√
user group	X	X	X	X	√

these datasets cover a broad spectrum in terms of the user and item densities; that is, the volume of actions associated with a given user or item.

Amazon. Amazon.com furnished this study with the first batch of the large datasets, four of which were recently put forward by McAuley, Targett, Shi, and Van Den Hengel [30]. These massive datasets comprise time stamps and review texts that run from May 1996 to July 2014, and the authors of [30] have assembled every overarching product category on Amazon.com as separate datasets. For this study, a range of the major categories, such as automotive, cellphones and accessories, and clothing, were employed.

Epinions. For the second major dataset, the authors of [48] used Epinions.com, a widely used consumer-review website. Its inclusivity resembles the Amazon data, as it comprises all the actions of all the users between January 2001 and November 2013, thereby preserving the sequential relationships.

Foursquare. The next dataset was derived from Foursquare.com, and this is a dataset is frequently used for gauging the performances of the next point-of-interest prediction algorithms. This dataset encompasses a sizeable volume of the user check-ins (or visits) for a variety of venues, such as shops and restaurants, between December 2011 and April 2012.

In all these datasets, the inactive users and any item lacking more than five associated actions were eliminated. The star-ratings were translated into tacit feedback (or “binary” actions) through the setting of the associated entries as 1; that is, the authors’ interest reflects the actions (purchases, reviews, check-ins, etc.) without a consideration of the given rating values. Table 5 presents the postprocessing statistics of each dataset.

4.2. Baseline methods

We employ several state-of-the-art approaches with regard to sequential prediction and item recommendation (see Table 6).

(1) Bayesian personalized ranking (BPR-MF) [39] is a top-of-the-line approach for individualized item recommendation. Employing matrix factorization [23] as the fundamental predictor, it gauges only long-term preferences.
(2) Factored item similarity models (FISM) [18] is a cutting-edge similarity-based algorithm, also used for individualized item recommendations. Our model improves upon it to address the issue of sequential prediction.

(3) Factorized personalized Markov chain (FPMC) [40] is an approach employing a personalized Markov chain in order to make sequential predictions.
(4) Factorized sequential prediction with item similarity models (Fossil) [13] is an algorithm that makes sequential predictions by employing the item-to-item similarity method and high-ordered Markov chains.
(5) Factorized group preference-based similarity models for sparse sequential recommendation (GPS) is the algorithm we propose in this study. In it, we include group preference and combined different-order Markov chains.

4.3. Evaluation methodology

We seek to ascertain the efficacy of our approach by studying first the short-term and then the long-term predictions (that is, the first item rated and then all the subsequent items rated within the test period). The evaluator metrics we use are short-term prediction success (SPS), recall, and NDCG (normalized discounted cumulative gain):

Short-term prediction success, used by many researchers (e.g., [9,33]), represents a method's capability of forecasting the next item—1 if that item is included in the recommendations, 0 if it is not. We calculate SPS as shown in Eq. (16):

$$\text{sps@k} = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} 1\left(S_1^u \in T^u\right) \tag{16}$$

where T^u is the recommended top-k items for user u. The indicator function 1(b) returns 1 if the argument b is true and 0 otherwise.

The standard metric for the top-N recommendation portrays a method's capability for long-term predictions, as shown in Eq. (17):

$$\text{recall@k} = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} \frac{|GT^u \cap T|}{|GT^u|} \tag{17}$$

where GT^u is the ground-truth item of user u.

To measure the performances of recommender algorithms, several researchers adopted NDCG, which attempts to measure the rank performance between predicted ratings and real values (e.g., [26, 29]). NDCG not only considers accuracy but also takes recommendation order into account. DCG is defined in Eq. (18):

$$\text{DCG@k} = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\sum_{i=1}^{k}\left(\frac{1\left(T_i^u \in GT^u\right)}{\log_2(2+i)}\right) \tag{18}$$

NDCG@k is the DCG@k normalized to [0, 1], where one signifies a perfect ranking.

These metrics are calculated "at 30"—that is, in a given context in which the recommendation system generates thirty recommendations per user.

4.4. Statistical significance

In this section, to show that the observed differences in our experimental results using SPS, recall, and NDCG are not incidental, we provide a significance test. Two of the most commonly used significance tests used in information-retrieval experiments are the Wilcoxon signed-rank test and the *t*-test [41]. In short, both take a pair of equal-sized sets of per-query effectiveness values and assign a confidence value to the null hypothesis that the values are drawn from the same distribution. The results of the experiments are reliable and convincing if confidence in the hypothesis (reported as a p-value) is less than 5%.

The Wilcoxon signed-rank test and the *t*-test assume that the values being tested are distributed symmetrically and normally, respectively [41]. However, effectiveness rarely follows either distribution. Instead, Hull [16] points out that the *t*-test can be reliable even when the data being tested are not distributed normally. Therefore, we applied a paired *t*-test to determine whether the observed difference is incidental.

4.5. Overall performance

A paired *t*-test shows that in using sps@30, recall@30, and ndcg@30 as performance measures, our approach performs significantly better than the baseline methods as shown in Table 7 at p=0.05, p=0.025, and p=0.05, respectively. In the table, the p values of these tests are all less than 0.05, which means that the results of the experiments are statistically significant. We believe that such gains are introduced by the introduction of group preference.

5. Results

We assess the results of this experiment in four stages. In Section 5.1, we establish the comparative outcomes of the different approaches for every dataset. In assessing divergent datasets, evaluation results are depicted for various degrees of scarcity. Essentially, the learning rate = 0.02, K = 100, H = 30, the representative datasets number 30, and the clusters

Table 7
A paired *t*-test of various methods.

	sps@30	recall@30	ndcg@30
Methods	**p-value**	**p-value**	**p-value**
BPRMF	0.000	0.000	0.000
FISM	0.004	0.020	0.039
FPMC	0.000	0.000	0.000
Fossil	0.025	0.005	0.003

Table 8
Results (SPS).

Datasets	method	BPR-MF (a)	FISM (b)	FPMC (c)	Fossil (d)	GPS (e)	improvement			
							d vs a	e vs b	e vs d	e vs best
A-Auto	sps@30	0.0384	0.0882	0.0275	0.0863	0.1012	0.048	0.013	0.015	0.013
A-Video	sps@30	0.0327	0.1072	0.0399	0.0875	0.1493	0.055	0.042	0.062	0.042
A-Elec	sps@30	0.0411	0.0421	0.0309	0.0428	0.0511	0.002	0.009	0.008	0.008
A-Office	sps@30	0.0386	0.1003	0.0630	0.1390	0.1461	0.100	0.046	0.007	0.007
Epinions	sps@30	0.1184	0.1147	0.0789	0.1184	0.1974	0.000	0.083	0.079	0.079
Foursquare	sps@30	0.2555	0.2622	0.2516	0.3162	0.3262	0.061	0.064	0.010	0.010
avg.(k=100)	sps@30	0.0919	0.1185	0.0815	0.1298	0.1669	0.038	0.048	0.037	0.034

Table 9
Results (Recall).

Datasets	method	BPR-MF (a)	FISM (b)	FPMC (c)	Fossil (d)	GPS (e)	Improvement			
							d vs a	e vs b	e vs d	e vs best
A-Auto	recall@30	0.0386	0.0834	0.0263	0.0821	0.0954	0.044	0.012	0.013	0.012
A-Video	recall@30	0.0334	0.1009	0.0387	0.0831	0.1456	0.050	0.045	0.063	0.045
A-Elec	recall@30	0.0436	0.0437	0.0309	0.0442	0.0509	0.001	0.007	0.007	0.007
A-Office	recall@30	0.0380	0.0756	0.0436	0.0750	0.0830	0.037	0.007	0.008	0.007
Epinions	recall@30	0.0727	0.0902	0.0370	0.0848	0.1390	0.012	0.049	0.054	0.049
Foursquare	recall@30	0.2382	0.221	0.2314	0.2517	0.2634	0.014	0.042	0.012	0.012
avg.(k=100)	recall@30	0.0767	0.1007	0.0636	0.1010	0.1309	0.024	0.030	0.030	0.026

Table 10
Results (NDCG).

Datasets	method	BPR-MF (a)	FISM (b)	FPMC (c)	Fossil (d)	GPS (e)	Improvement			
							d vs a	e vs b	e vs d	e vs best
A-Auto	ndcg @30	0.0169	0.0479	0.0136	0.0397	0.0504	0.0228	0.0025	0.0107	0.0025
A-Video	ndcg @30	0.0292	0.0830	0.0321	0.0679	0.0888	0.0387	0.0058	0.0209	0.0058
A-Elec	ndcg @30	0.0262	0.0265	0.0189	0.0268	0.0402	0.0006	0.0137	0.0134	0.0134
A-Office	ndcg @30	0.0202	0.0498	0.0237	0.0456	0.0549	0.0254	0.0051	0.0093	0.0051
Epinions	ndcg @30	0.0727	0.0902	0.0370	0.0848	0.1390	0.0121	0.0488	0.0542	0.0488
Foursquare	ndcg @30	0.1367	0.1399	0.1294	0.1589	0.1973	0.0222	0.0574	0.0384	0.0384
avg.(k=100)	ndcg @30	0.0503	0.0729	0.0425	0.0706	0.0951	0.0203	0.0222	0.0245	0.0190

number 30. In the second stage, we assess the effect of parameter values—thus, the quantity of representative datasets and the quantity of clusters. With regard to this section, please note that the Epinions dataset has been studied; however, similar results and conclusions can be drawn from the other datasets (see Section 5.2). Third, we compare the performance of GPS regarding data sparsity (see Section 5.3). Finally, in Section 5.4, we offer visualization and qualitative analysis of the GPS method.

5.1. Performance and analysis

We assess all approaches for predictive accuracy on the same terms (Section 4.3). To meaningfully compare the accuracy between different approaches, we employ the same dimensions ($K = 100$) with various values. Tables 8, 9, and 10 provide the results produced by SPS, recall, and NDCG across all datasets.

FISM vs. BPR-MF. Both BPR-MF and FISM offer strong approaches for modeling the preferences of individual users over time. While both methods factorize a matrix, they depart from each other in both their rationales and performances. BPR-MF depends on factorizing the user-item interaction matrix and deploying a K-dimensional vector to establish parameters

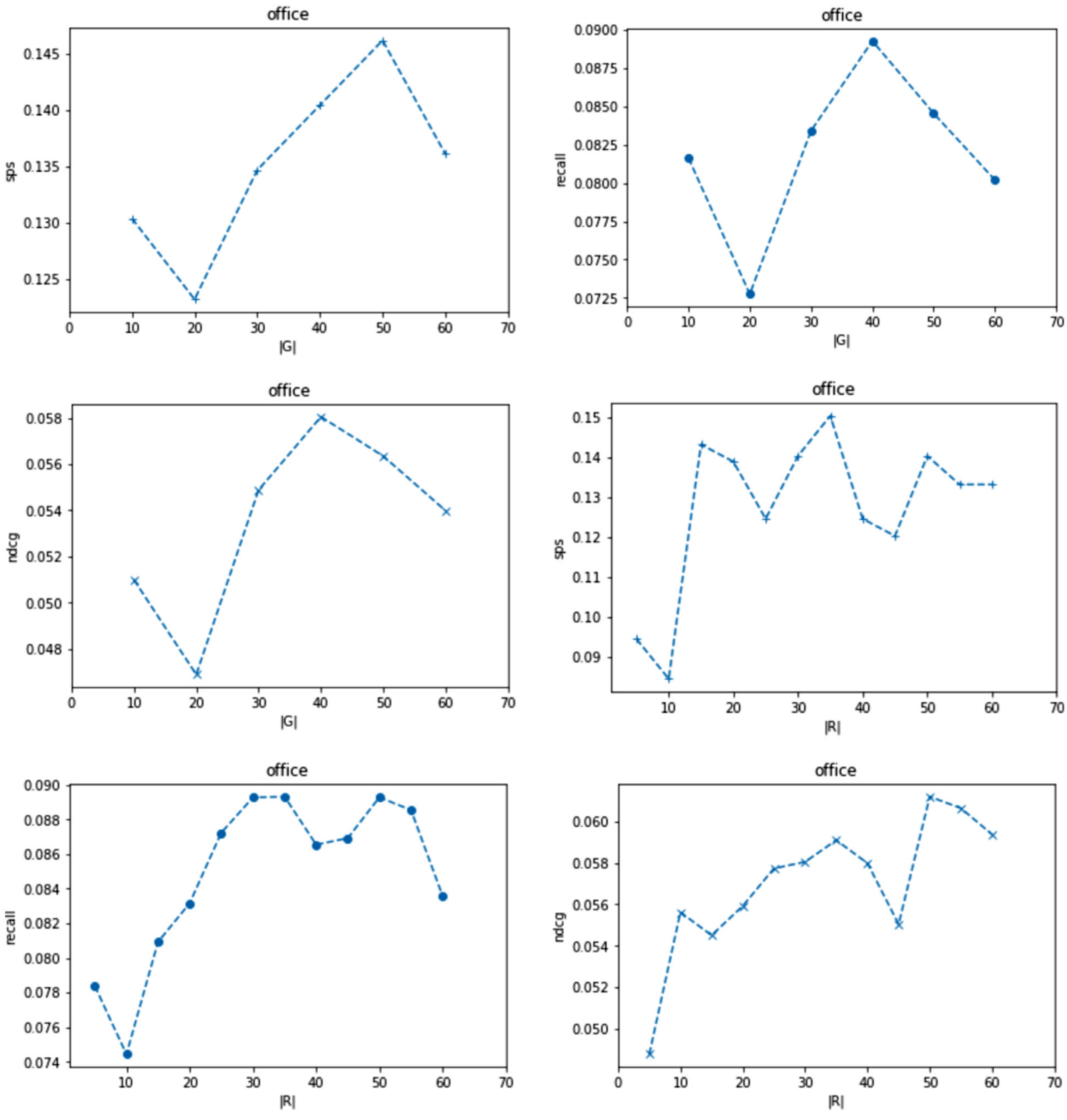

Fig. 2. Recommendation performances of GPS with different sizes of user groups and representative item set sizes.

Table 11
MovieLens Datasets.

Dataset	Threshold	#users	#items	#actions	avg.#actions/user	avg.#actions/item
ML-50	50	6040	2909	214,412	35.49	73.70
ML-30	30	6040	2711	150,612	24.93	55.55
ML-20	20	6040	2527	109,282	18.09	43.24
ML-10	10	6040	2122	57,445	9.5	27.07

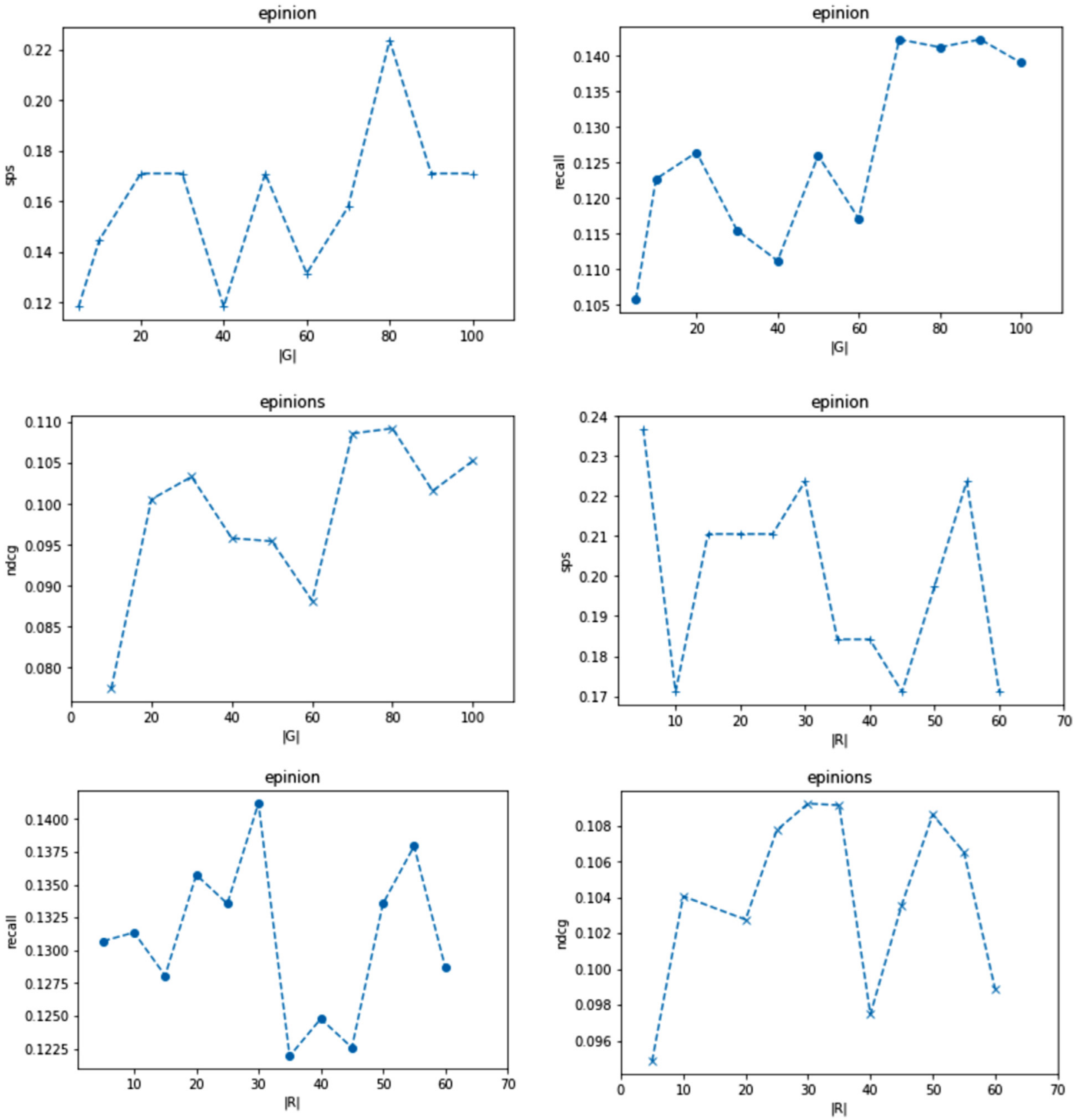

Fig. 2. Continued

for each user. FISM, on the other hand, factorizes the item-item similarity matrix, thereby diminishing the importance of establishing parameters for each user (who perhaps has only a few actions logged in the training set). This seeming advantage is borne out in our experiments, the results of which demonstrate FISM as notably better than BPR-MF on all datasets (typically above 0.0274). As such, FISM is a robust stepping-stone for our challenge of the similarity model.

Fossil vs. FPMC. Fossil promises to improve on FISM—up to 2.21 percent typically—because it combines FISM's robust modeling of long-term dynamics from limited data with Markov chains. When we assessed Fossil against FPMC, we learned that Fossil outperforms FPMC by a sizeable margin across the board (an average improvement of 0.0379). Fossil thus proves its ability to tackle real-world datasets of varying sizes.

GPS vs. FOSSIL. In conjoining FISM (which excels in producing models of long-term dynamics from limited data) with group preference and Markov chains, GPS improves on Fossil's results by an average of up to 0.0305. In assessing GPS against Fossil, we observed several distinctions: First, GPS outperforms Fossil across all datasets by an average of 0.0305;

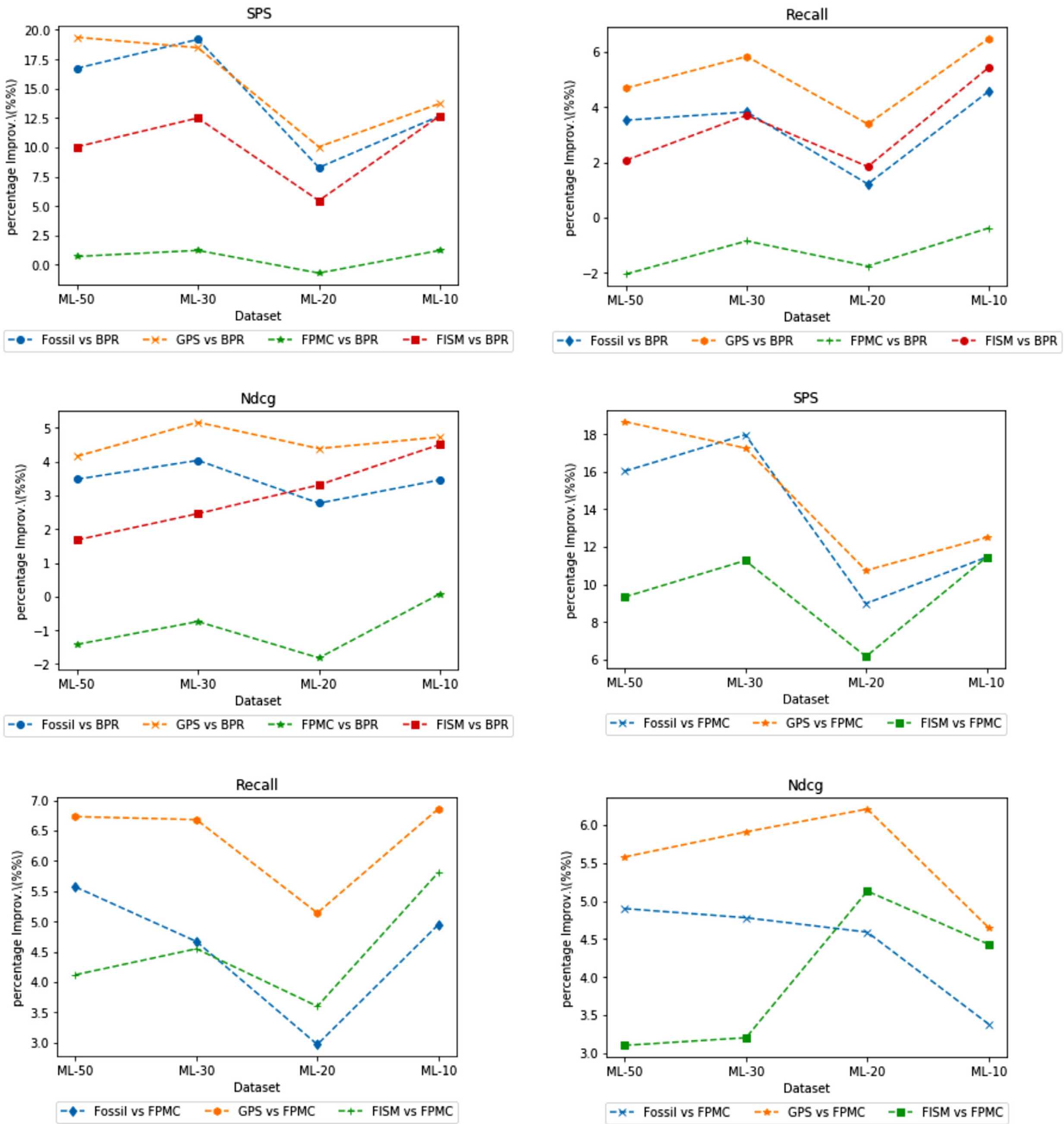

Fig. 3. Effect of group preference on data sparsity (comparison of GPS, Fossil, FPMC, FISM, and BPR-MF).

and second, it performs even better on sparse datasets (e.g., Epinions). This across-the-board robust performance attests to this model's capabilities in regard to tackling real-world datasets.

GPS vs. the best baselines. *The* right-hand column of Tables 8–10 shows the degree to which GPS surpasses the top baseline approaches for each case. Our study shows that GPS outperforms all baselines across the board, with an average improvement of 0.0263.

5.2. Impact of group and representative item set size

To fully grasp the effect of group pairwise preference in GPS, we modify group size as |G|={10,20,30,40,50,60} or |G|={10,20,30,40,50,60,70,80,90,100} and in Fig. 2, provide the outcomes of SPS@30, Recall@30, and NDCG@30.

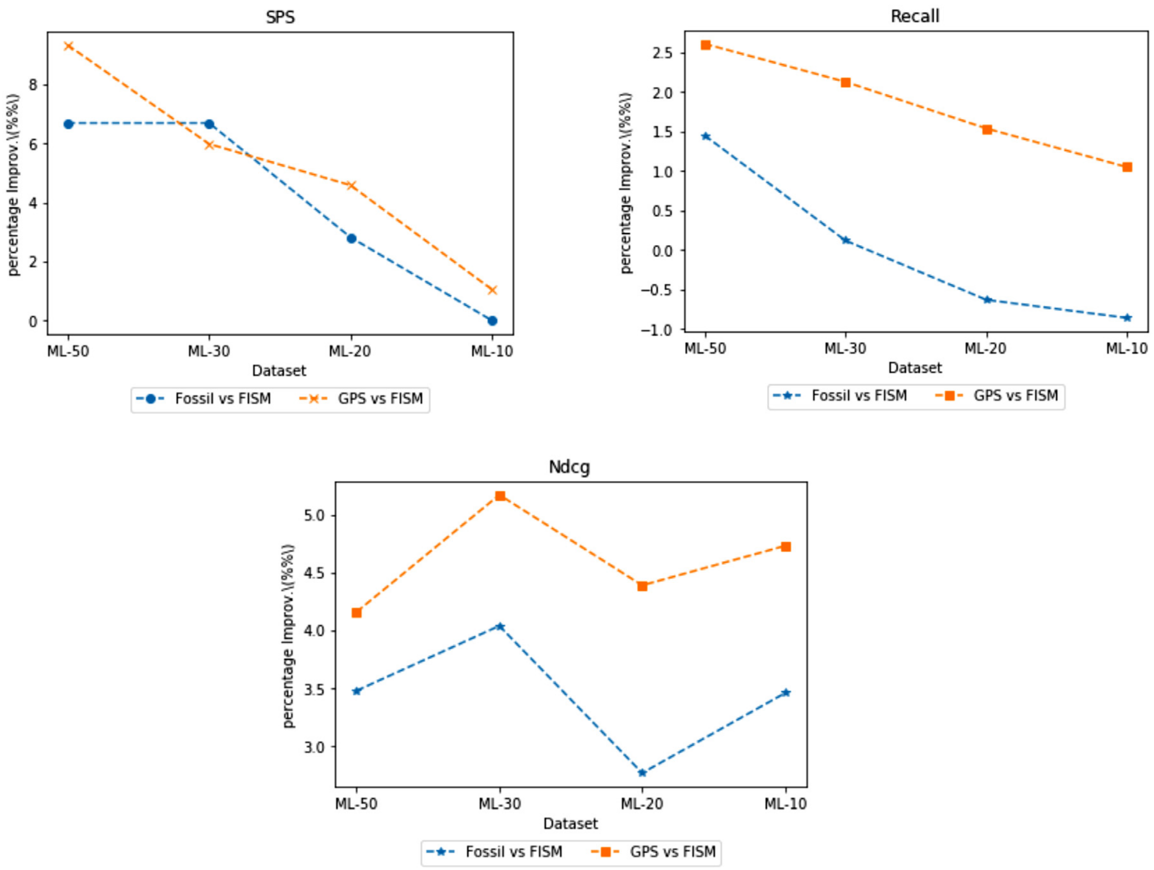

Fig. 3. Continued

As Fig. 2 demonstrates, the recommendation performance markedly improves on all datasets when one increases the size of the user group (e.g., |G|=40 or 80). This pattern is attributable to the impact of modeling group preference in Eq. (12) and grasping model parameters in Eq. (13) by adding the user group G. To better understand the results of the inserted interactions by way of user groups and representative item sets, we expand the empirical investigation by using the better-performing group with a new representative item set, i.e., |R|={5,10,15,20,25,30,35,40,45,50,55,60}.

Fig. 4 (presenting the recommendation results and observations) reveals the usefulness of inserting such user group and/or representative item interactions into GPS to augment sequential prediction above a primary baseline (BPR-MF).

5.3. The effect of data sparsity

We subsequently consider the impact that dataset sparsity has on the various techniques by experimenting on a commonly used dataset, MovieLens-1 M. This densely populated dataset is comprised of approximately one million user ratings, with 6040 users rating 3706 movies between April 2000 and February 2003. Here, as before, star-ratings are translated as implicit feedback for the purpose of our study, and K dimensions is set at 100.

We construct a dataset sequence in which each dataset has a distinct threshold N, where N indicates the number of the most recent actions for the given user that are included (and any actions past N are omitted). We do not employ sampling, as it would violate precisely the sequentiality that the models rely on. By reducing the N-threshold from 50 down to 10, we are able to see how the performance of each method varies as the datasets become more and more sparse. Table 11 presents the dataset statistics, and Table 12 collates the outcomes of the experiment. Table 12 demonstrates clearly that most techniques experience diminished accuracy as the threshold also decreases—which is intuitive, given that these datasets offer less information concerning users and item-to-item transitions. Here, we assess GPS against Fossil to gain insight into matters pertaining to their efficacy, based on the outcomes presented in Table 12.

5.3.1. The effect of group preference on data sparsity

Concerning the impact of group preference on data sparsity, we first consider BPR-MF against other methods. As Fig. 3 demonstrates, GPS outperforms all other methods with respect to BPR-MF. We likewise consider the advantages of

Fig. 4. Visualization of GPS recommendation.

Table 12
Results (SPS and Recall) of MovieLens Datasets.

Dataset	Matric	BPR-MF	FISM	FPMC	Fossil	GPS
ML-50	Spsrecall	0.11610.0977	0.21650.1185	0.12320.0773	0.28340.1330	0.30980.1446
	Ndcg	0.0783	0.0951	0.0641	0.1131	0.1199
ML-30	Spsrecall	0.11090.0939	0.23590.1310	0.12320.0855	0.30280.1322	0.29570.1523
	Ndcg	0.0731	0.0977	0.0657	0.1135	0.1248
ML-20	Spsrecall	0.13550.1002	0.19010.1187	0.12850.0827	0.21830.1124	0.23590.1341
	Ndcg	0.0709	0.1040	0.0527	0.0986	0.1148
ML-10	Spsrecall	0.08620.0914	0.21300.1458	0.09850.0877	0.21300.1372	0.22350.1563
	Ndcg	0.0543	0.0994	0.0551	0.0889	0.1016

FPMC and FISM against other techniques, both of which illustrate the superiority of GPS over all other techniques. Although it apparently helps most in cases where the dataset is well-populated (e.g., ML-50 vs. others), group preference demonstrably imparted benefits even in cases of data sparsity.

5.4. Visualization of the recommendations

We show in Fig. 4 some of the recommendations furnished by GPS (K=100) within the dataset of Electronics. Sampling users from the database at random, we present their historical sequences in the left-hand column, their top-five recommendations in the central column, and the ground-truth on the right-hand side. The upshot shows how GPS ably grasps products closely related to the ground-truth—as in the case when it recommends photography-related items aligned with the ground truth for users one, two, and five (all of whom are apparently photographers). GPS also furnishes suggestions for computer-related products that align with user three's purchase history. Likewise, it recommends sonic or musical products for user six, who evinces musical interest.

6. Conclusions

We have put forth in this paper a factorized group preference-based similarity (or GPS) that combines similarity-based techniques with Markov chains to forecast individualized sequential behavior. It thus employs methods of similarity to recommend items that are not sequentially aware, and at the same time makes use of Markov chains to individualize sequential recommendations. GPS likewise makes use of principles of group preference in addition to user preference, thereby imparting deeper layers of interactions among users. This has the great advantage of minimizing the problems of data sparsity or cold users, and furthermore eliminates some strong independency assumptions concerning the different factors. By running a range of qualitative and quantitative experiments on a range of sizeable real-world datasets, we demonstrate that GPS outperforms a number of top-of-the-line approaches—particularly with regard to sparse datasets. GPS likewise proves itself capable of ascertaining individualized dynamics and generating strong recommendations. These empirical results lend support to one of our governing presuppositions: that introducing notions of group preference into instances of user preference leads to a higher accuracy in recommendations.

Moving forward, our intention is to experiment with GPS on a range of sizeable educational datasets, with an eye towards building a system that offers adaptive educational recommendations. Furthermore, we hope to further develop this model through additional user- and item-feature information, which could continue to ameliorate the challenges associated with cold-start problems and data sparsity.

Acknowledgments

Special thanks to the anonymous reviewers for their insightful comments which helped us make this paper better. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP)[grant number NRF-2016R1A2B2015912] and Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT)(No. 2016-0-00010-003, Digital Content In-House R&D).

References

[1] G. Adomavicius, 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. 6 (2005) 734–749.
[2] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Mach. Learn. 6 (1991) 37–66.
[3] A. Bellogín, P. Sánchez, Collaborative filtering based on subsequence matching: a new approach, Inf. Sci. 418 (2017) 432–446.
[4] J.S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc., 1998, pp. 43–52.
[5] C. Cai, R. He, J. McAuley, SPMC: socially-aware personalized markov chains for sparse sequential recommendation, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, AAAI Press, 2017, pp. 1476–1482.
[6] A. Calma, T. Reitmaier, B. Sick, Semi-supervised active learning for support vector machines: A novel approach that exploits structure information in data, Inf. Sci. 456 (2018) 13–33.
[7] Y. Cheng, L. Yin, Y. Yu, LorSLIM: low rank sparse linear methods for top-n recommendations, in: Data Mining (ICDM), 2014 IEEE International Conference on, IEEE, 2014, pp. 90–99.
[8] K.J. Cios, W. Pedrycz, R.W. Swiniarski, L.A. Kurgan, Data mining: a Knowledge Discovery Approach, Springer Science & Business Media, 2007.
[9] R. Devooght, H. Bersini, Long and short-term recommendations with recurrent neural networks, in: Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, ACM, 2017, pp. 13–21.
[10] R. Francesco, R. Lior, S. Bracha, Introduction to recommender systems handbook, RecommenderSystems Handbook, Springer, Amerika Serikat, 2011 in.
[11] K. Goldberg, T. Roeder, D. Gupta, C. Perkins, Eigentaste: a constant time collaborative filtering algorithm, Inf. Retr. 4 (2001) 133–151.
[12] G. Guo, F. Zhu, S. Qu, X. Wang, PCCF: Periodic and continual temporal co-factorization for recommender systems, Inf. Sci. 436 (2018) 56–73.
[13] R. He, J. McAuley, Fusing similarity models with markov chains for sparse sequential recommendation, in: Data Mining (ICDM), 2016 IEEE 16th International Conference on, IEEE, 2016, pp. 191–200.
[14] T.V. Himabindu, V. Padmanabhan, A.K. Pujari, Conformal matrix factorization based recommender system, Inf. Sci. (2018).
[15] Y. Hu, Y. Koren, C. Volinsky, Collaborative filtering for implicit feedback datasets, in: Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on, Ieee, 2008, pp. 263–272.
[16] D. Hull, Using statistical testing in the evaluation of retrieval experiments, in: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 1993, pp. 329–338.
[17] K.-S. Hwang, Y.-J. Chen, W.-C. Jiang, T.-W. Yang, Induced states in a decision tree constructed by Q-learning, Inf. Sci. 213 (2012) 39–49.
[18] S. Kabbur, X. Ning, G. Karypis, Fism: factored item similarity models for top-n recommender systems, in: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2013, pp. 659–667.
[19] M. Kárný, Recursive estimation of high-order Markov chains: approximation by finite mixtures, Inf. Sci. 326 (2016) 188–201.
[20] K.-J. Kim, H. Ahn, A recommender system using GA K-means clustering in an online shopping market, Expert Syst. Appl. 34 (2008) 1200–1209.
[21] Y. Koren, Collaborative filtering with temporal dynamics, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2009, pp. 447–456.
[22] Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2008, pp. 426–434.
[23] Y. Koren, R. Bell, in: Advances in Collaborative filtering, in: Recommender systems Handbook, Springer, 2015, pp. 77–118.
[24] V. Kumar, A.K. Pujari, S.K. Sahu, V.R. Kagita, V. Padmanabhan, Collaborative filtering using multiple binary maximum margin matrix factorizations, Inf. Sci. 380 (2017) 1–11.
[25] J.-S. Lee, C.-H. Jun, J. Lee, S. Kim, Classification-based collaborative filtering using market basket data, Expert Syst. Appl. 29 (2005) 700–704.
[26] D. Liang, J. Altosaar, L. Charlin, D.M. Blei, Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence, in: Proceedings of the 10th ACM conference on recommender systems, ACM, 2016, pp. 59–66.
[27] M.M. Lin, B. Dong, M.T. Chu, Integer matrix factorization and its application, Tech. Rep., (2005).
[28] G. Linden, B. Smith, J. York, Amazon. com recommendations: Item-to-item collaborative filtering, IEEE Internet Comput. (2003) 76–80.
[29] Y. Liu, J. Yang, Improving ranking-based recommendation by social information and negative similarity, Procedia Comput. Sci. 55 (2015) 732–740.
[30] J. McAuley, C. Targett, Q. Shi, A. Van Den Hengel, Image-based recommendations on styles and substitutes, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2015, pp. 43–52.

[31] A. Mnih, R.R. Salakhutdinov, Probabilistic matrix factorization, in: Advances in Neural Information Processing Systems, 2008, pp. 1257–1264.
[32] X. Ning, G. Karypis, Slim: Sparse linear methods for top-n recommender systems, in: 2011 11th IEEE International Conference on Data Mining, IEEE, 2011, pp. 497–506.
[33] S. Niu, R. Zhang, Collaborative sequence prediction for sequential recommender, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, ACM, 2017, pp. 2239–2242.
[34] F. Ortega, A. Hernando, J. Bobadilla, J.H. Kang, Recommending items to group of users using matrix factorization based collaborative filtering, Inf. Sci. 345 (2016) 313–324.
[35] R. Pan, M. Scholz, Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2009, pp. 667–676.
[36] W. Pan, L. Chen, in: GBPR: Group Preference Based Bayesian Personalized Ranking For One-Class Collaborative Filtering, IJCAI, 2013, pp. 2691–2697.
[37] W. Pan, L. Chen, Group Bayesian personalized ranking with rich interactions for one-class collaborative filtering, Neurocomputing 207 (2016) 501–510.
[38] H. Qiu, Y. Liu, G. Guo, Z. Sun, J. Zhang, H.T. Nguyen, BPRH: Bayesian personalized ranking for heterogeneous implicit feedback, Inf. Sci. 453 (2018) 80–98.
[39] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, in: Proceedings of the twenty–fifth conference on uncertainty in artificial intelligence, AUAI Press, 2009, pp. 452–461.
[40] S. Rendle, C. Freudenthaler, L. Schmidt-Thieme, Factorizing personalized markov chains for next-basket recommendation, in: Proceedings of the 19th international conference on World wide web, ACM, 2010, pp. 811–820.
[41] M. Sanderson, J. Zobel, Information retrieval system evaluation: effort, sensitivity, and reliability, in: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2005, pp. 162–169.
[42] S. Sedhain, A.K. Menon, S. Sanner, L. Xie, D. Braziunas, in: Low-Rank Linear Cold-Start Recommendation from Social Data, AAAI, 2017, pp. 1502–1508.
[43] G. Shani, D. Heckerman, R.I. Brafman, An MDP-based recommender system, J. Mach. Learn. Res. 6 (2005) 1265–1295.
[44] V.A. Shim, K.C. Tan, C.Y. Cheong, J.Y. Chia, Enhancing the scalability of multi-objective optimization via restricted Boltzmann machine-based estimation of distribution algorithm, Inf. Sci. 248 (2013) 191–213.
[45] X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques, Advan. Artif. intell. 2009 (2009).
[46] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, X. Cheng, Learning hierarchical representation model for nextbasket recommendation, in: Proceedings of the 38th International ACM SIGIR conference on Research and Development in Information Retrieval, ACM, 2015, pp. 403–412.
[47] Y. Wang, J. Deng, J. Gao, P. Zhang, A hybrid user similarity model for collaborative filtering, Inf. Sci. 418 (2017) 102–118.
[48] T. Zhao, J. McAuley, I. King, Leveraging social connections to improve personalized ranking for collaborative filtering, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, ACM, 2014, pp. 261–270.
[49] A. Zimdars, D.M. Chickering, C. Meek, Using temporal data for making recommendations, in: Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc., 2001, pp. 580–588.