The Journal of Supercomputing (2021) 77:7509–7534
https://doi.org/10.1007/s11227-020-03555-w

Sequence-aware similarity learning for next-item recommendation

Zhuoxin Zhan[1] · Liulan Zhong[1] · Jing Lin[1] · Weike Pan[1] · Zhong Ming[1]

Accepted: 30 November 2020 / Published online: 4 January 2021

Abstract
Sequence-aware next-item recommendation has recently been studied because of the noteworthy usefulness of the sequential information integrated into recommendation algorithms. Following the development thread of sequential recommendation methods, especially the factored Markov chains segment, and based on a well-designed fusing similarity model with factored high-order Markov chains (i.e., Fossil), we propose a novel and generic similarity learning framework for next-item recommendation called sequence-aware factored mixed similarity model (S-FMSM), which contains two variants with pairwise preference learning and pointwise preference learning. Unlike the baseline methods that model the general representation and the sequential representation in two divided factorization components, we use a factored mixed similarity model that unites the general similarity and the sequential relationship between two successive items for their sequential representation learning. Experiments on six datasets show that our newly introduced general similarity can notably improve the results of the recommended ranking lists. Furthermore, a study on tuning the prior trade-off parameter indicates the importance of the general similarity on different datasets. We adjust the number of latent dimensions and try different similarity measurements, which showcases that our S-FMSM is universal and gives us more insight on it.

Keywords Sequential recommendation · Factored Markov chains · Factored mixed similarity model

This work is an extension of our previous work [32]. Compared with our previous work, we have added the following new contents in this paper, (i) we have developed an extension of S-FMSM, i.e., a pointwise preference learning method S-FMSM(poi), in Sect. 6; (ii) we have included more empirical results, i.e., an additional baseline method TransRec in Table 3, Fig. 5, and Table 5, results of methods with different numbers of latent dimensions in Fig. 7, and results with different similarity measurements in Table 6, as well as the corresponding descriptions and discussions; (iii) we have included more related works in Sect. 2; and (iv) we have made many improvements throughout the whole paper, including figure illustrations, and more discussions and analysis.

Zhuoxin Zhan and Liulan Zhong are co-first authors.

Extended author information available on the last page of the article

1 Introduction

Under the trend of information sharing and goods globalization, finding the items that really interest users becomes challenging because of the information overload effect, for which recommender systems have emerged in various scenarios such as online entertainment and commercial platforms [12]. A typical recommender system is generally composed of three components, i.e., input data collected from users' behavior information, recommendation algorithms for users' preference learning, and systems for user–item interactions. Among the above three components, recommendation algorithms are the hard core of recommender systems and are extensively studied. While for different applications, the goals of recommendation are generally different. In early times, straightforward recommendation can be abstracted as a one-class collaborative filtering (OCCF) problem, in which only the user- and item-IDs of positive feedback are considered. Methods for the OCCF problem can be roughly divided into neighborhood-based methods [1] and model-based methods [15, 22]. With the explosive increase of the item and user amount, neighborhood-based methods become more and more complexity unaffordable on both time and memory. Model-based models address this issue well and therefore become research hotspots in academia and industry. Factored item similarity models (FISM) [9] firstly introduces an item-to-item similarity matrix on the basis of matrix factorization (MF) and obtain much better recommendation performance compared with other MF-based models. Lately, it is found that sequential information can be included to specify the problem into next-item recommendation [21] (see Fig. 1).

Although sequential recommendation can be achieved by general recommendation methods [9, 19, 23, 25], the shortcomings of these algorithms lie in insufficient use of the available information, unawareness of the users' preference dynamics, and neglect of the structural relationship among items. To manage the sequential information in recommender systems, there are some works that model users' sequential feedback as Markov chains (MC). As shown in Fig. 2, Markov chains make use of the information of users' preceding interacted items and thus address the sequential recommendation problem.

Methods for sequential recommendation can be divided into purely MC-based models [33], factored MC models with users' global preferences [3, 24], other rising methods such as translation-based models [5, 18], and deep learning-based models [8, 11, 27–29]. [24] firstly combines matrix factorization and (first-order) Markov

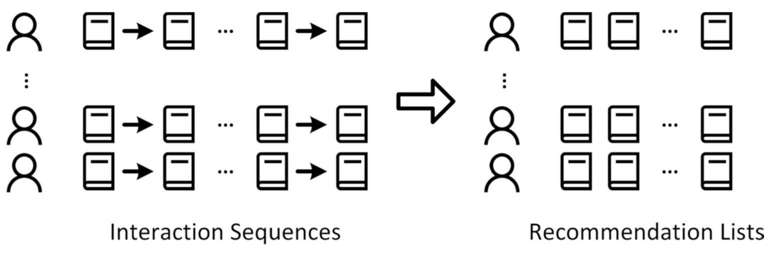

Fig. 1 Illustration of the studied next-item recommendation problem, where we exploit interaction sequences of users to items so as to provide a personalized recommendation list for each user

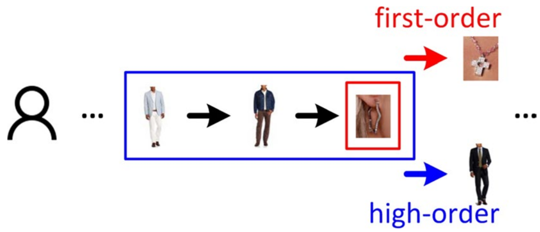

Fig. 2 Illustration of first- and high-order Markov chains

chains together to address the sequential recommendation problem. [3] improves the MF and first-order MC to the similarity-based method FISM and high-order MC. It introduces more item–item and user–item interactions, and therefore, successfully addresses the sparsity issues and further improves the recommendation performance.

Through a deep investigation of the factored MC models, we focus on the improvement of Fossil [3]; a well-rounded model not only takes an item similarity matrix for users' general representation learning, but also assigns some personalized and order-sensitive weights for sequential representation learning. Different from the methods mentioned above, our proposed model, sequence-aware factored mixed similarity model (or S-FMSM for short) integrates items' general similarity and items' learnable sequential representations in a unified component through a factored mixed similarity model [13]. Fossil is a representative work in the area of sequential recommendation. We introduce an important concept in collaborative recommendation, i.e., mixed similarity, into the modeling of sequential feedback, which is a significant extension of Fossil. Specifically, we apply the general similarity (i.e., cosine similarity and Jaccard index) between two successive items as a prior coefficient of the corresponding factored sequential interaction. Following two preference learning paradigms, we then design two variants of S-FMSM, i.e., S-FMSM(pai) and S-FMSM(poi).

Extensive empirical studies on six public datasets show that our S-FMSM outperforms all the factorization-based baselines on almost all the evaluation metrics. We also conduct some studies of the trade-off parameter to verify the importance of our newly introduced similarity prior on different datasets.

Our main contributions are as follows: (i) we firstly introduce a mixed similarity into the studied sequential recommendation problem, which can be seamlessly combined with a factorization-based sequential recommendation method to obtain effective and consistent improvement; (ii) we design a novel algorithm termed S-FMSM with two variants, by combining the mixed similarity with a well-rounded model (i.e., Fossil); (iii) we conduct extensive experiments on six datasets, which showcase that our proposed S-FMSM outperforms the state-of-the-art methods; and (iv) we study the effect of the trade-off parameter in the mixed similarity, the number of latent dimensions and different similarity measurements, which give us more insight on the proposed model.

The rest of the paper is organized as follows. We first discuss the related work in Sect. 2. Then we discuss the background and our preliminaries in Sect. 3. After that, we propose our model in Sect. 4, and further design two variants, i.e., S-FMSM(pai) and S-FMSM(poi), in Sects. 5 and 6, respectively. We conduct extensive experiments and analyze the results in Sect. 7. Finally, we conclude our work in Sect. 8.

2 Related work

2.1 General recommendation

The widespread use of recommender systems has confirmed the successful development of recommendation algorithms. The most conventional and state-of-the-art algorithms model users' preferences by utilizing the simplest form of users' historical feedback only. Specifically, with implicit feedback (i.e., non-scoring actions such as clicks, likes and purchases), each record in the dataset can be denoted as a (user, item) pair, in which only user IDs and item IDs are left. In this case, to serve a specific user, it is proper to consider all his/her interacted items equally, which is therefore called general recommendation.

There are many branches of general recommendation methods. For example, neighborhood-based methods [25] directly calculate the similarity between two users/items through a similarity measurement such as cosine similarity and then obtain the preference score with a weighted function. To be specific, the cosine similarity between item i and item j is computed by the following formula:

$$s_{ij} = \frac{|\mathcal{U}_i \cap \mathcal{U}_j|}{\sqrt{|\mathcal{U}_i|}\sqrt{|\mathcal{U}_j|}}, \tag{1}$$

where $\mathcal{U}_i$ and $\mathcal{U}_j$ are sets of users that have interacted with item i and item j, respectively. After that, the preference score of user u to item i is obtained by:

$$\hat{r}_{ui} = \sum_{k \in \mathcal{I}_u \cap \mathcal{N}_i} s_{ki}, \tag{2}$$

where $\mathcal{N}_i$ is the nearest neighbors of item i and $\mathcal{I}_u$ is the set of interacted items by user u. The neighborhood-based methods make use of users' interaction information on items to calculate the item-to-item similarities, so as to make recommendations based on the assumption "a user will like some similar items to those he liked before".

Matrix factorization-based methods like RSVD [19] and its extensions [10, 17] learn a latent representation vector for each user and each item and then obtain the prediction score by the corresponding inner product. The prediction score of user u to item i by RSVD is as follows:

$$\hat{r}_{ui} = \mu + b_u + b_i + U_{u\cdot} V_{i\cdot}^T, \tag{3}$$

where $U_{u\cdot}$ and $V_{i\cdot}$ are the latent representation vectors of user u and item i, respectively. And μ, b_u, and b_i represent the global average score, user bias, and item bias, respectively. MF-based methods well address some limitations of memory-based methods such as non-transitivity and inefficiency.

Instead of learning the explicit user representations, a modified work of MF-based methods (i.e., FISM [9]) represents each user as a weighted sum of the representations of his/her interacted items, which are learned from another item-to-item similarity matrix. FISM represents the latent vector of user u w.r.t. item i as follows:

$$\bar{U}_{u\cdot}^{-i} = \frac{1}{\sqrt{|\mathcal{I}_u \backslash \{i\}|}} \sum_{i' \in \mathcal{I}_u \backslash \{i\}} W_{i'\cdot}, \tag{4}$$

where $W_{i'\cdot}$ is the auxiliary item-specific latent feature vector of item i' used to capture the item-to-item similarity. By applying the item-to-item similarity, FISM is able to cope with sparser data with fewer records from each user.

For more details of the implementation of MF methods, there are pointwise loss and pairwise loss [23], among which the pairwise one is often deemed more reasonable since it maximizes the prediction gap between positive records and negative ones. Recently, there are some deep learning-based methods such as NeuMF [7], which explores the nonlinearity of the latent representations. Furthermore, content-based recommendation [20] with more predefined features is also widely studied.

2.2 Sequential recommendation

To incorporate the sequential information of users' feedback into recommendation, Markov chains models are adopted [33]. Treating a user's interaction sequence as a sentence in a statistical language model, the probability of a sentence can be obtained based on the Bayesian formula, in which for a first-order Markov chain, the probability of the current item depends only on the probability of the former item.

Rendle et al. [24] propose FPMC to model Markov chains in a factorization way and combine it with the traditional MF for general representations of the users and items. According to FPMC, the predicted preference of user u to item i_u^t is as follows:

$$\hat{r}_{ui_u^t} = U_{u\cdot} V_{i_u^t\cdot}^T + P_{i_u^{t-1}\cdot} Q_{i_u^t\cdot}^T, \tag{5}$$

where $U_{u\cdot}$ and $V_{i\cdot}$ is the latent representation vector of user u and item i, respectively. $P_{i_u^{t-1}\cdot}$ and $Q_{i_u^t\cdot}$ is the auxiliary latent representation vector of item i_u^{t-1} and item i_u^t, respectively. FPMC captures both user's long-term and short-term preference via the first term and second term of the prediction rule in Eq. (5), respectively, so as to address the problem of sequential recommendation.

Similar to FISM, He et al. [3] propose Fossil to fuse similarity models into sequential recommendation and further elaborate more user- and order-sensitive parameters to handle personalized high-order Markov chains. The preference of user u to item i is estimated by:

$$\hat{r}_{ui_u^t} = b_{i_u^t} + \bar{U}_{u\cdot}^{-i_u^t} V_{i_u^t\cdot}^T, \tag{6}$$

where

$$\bar{U}_{u\cdot}^{-i_u^t} = \frac{1}{\sqrt{|\mathcal{I}_u \backslash \{i_u^t\}|}} \sum_{i' \in \mathcal{I}_u \backslash \{i_u^t\}} W_{i'\cdot} + \sum_{\ell=1}^{L} (\eta_\ell + \eta_\ell^u) W_{i_u^{t-\ell}\cdot}, \tag{7}$$

and η_ℓ^u controls the weight of user u's preference and sequential dynamics, while η_ℓ is a global parameter shared by all the users. By fusing similarity models and applying high-order Markov chains, Fossil is able to better address the sparsity issues and make more accurate personalized recommendation for users.

With longer subsequences to be considered, the research hot spot of sequential recommendation gradually shifts to the sequence modeling side. He et al. [5] model (user, item) interactions from a new perspective of translation and propose TransRec, whose prediction rule is as follows:

$$\hat{r}_{ui_u^t} = b_{i_u^t} - d(V_{i_u^{t-1}\cdot} + U_{u\cdot} + R_\cdot, V_{i_u^t\cdot}), \tag{8}$$

where $||V_{i_u^{t-1}\cdot}|| \le 1$, $||V_{i_u^t\cdot}|| \le 1$, $d(V_{i_u^{t-1}\cdot} + U_{u\cdot} + R_\cdot, V_{i_u^t\cdot}) = ||V_{i_u^{t-1}\cdot} + U_{u\cdot} + R_\cdot - V_{i_u^t\cdot}||$ is the Euclidean distance, and $R_\cdot$ is the global translation vector shared by all users.

However, our proposed model, which is based on the well-rounded Fossil, takes in a kind of more direct similarity prior knowledge (i.e., cosine similarity) to combine the general similarity and the sequential representations. This idea is expected to be helpful in other sequence modeling methods including deep learning-based approaches [8, 11, 27, 29] that we omit in this paper. Notice that we also omit those content-aware sequential recommendation methods [4, 26, 31] in this paper.

3 Background and preliminaries

3.1 Problem formulation

In one-class collaborative filtering with only positive feedback such as browses or clicks, given n users and m items, each user u is associated with a sequence of actions $\mathcal{S}_u = \{i_u^1, i_u^2, \dots, i_u^t, \dots, i_u^{|\mathcal{S}_u|}\}$, where i_u^t denotes the tth engaged item by user u. In this paper, we aim to present each user with a personalized ranking list of items at their next step. That is to say, our model takes a user's sequence as input $(u, \mathcal{S}_u)$ and ranks the unobserved items at the $(t+1)$th step by estimating the score $\hat{r}_{uj}, j \in \mathcal{I} \backslash \mathcal{I}_u$ to form the recommendation list (see Fig. 1). Some notations used in this paper and their explanations are shown in Table 1.

3.2 Challenges and overall of our solution

Our goal is to recommend a candidate list of items to each user u by exploiting the interaction data $(u, \mathcal{S}_u)$, for which we have to address the following challenges.

(1) The *sparsity* challenge. Due to the large amount of items in real-world datasets, the density of datasets is usually extremely low, which brings a lot of difficulties to our studied problem.

Table 1 Some notations and explanations commonly used in the paper

n	Number of users
m	Number of items
u	User ID, $u \in \{1, 2, \dots, n\}$
i	Item ID, $i \in \{1, 2, \dots, m\}$
$\mathcal{U}$	The whole set of users
$\mathcal{I}$	The whole set of items
$\mathcal{P}$	The whole set of observed (u, i) pairs
$\mathcal{A}$	A sampled set of unobserved (u, i) pairs
$\mathcal{I}_u$	A set of items that have been interacted by user u
$\mathcal{I}_u^{te}$	A set of preferred items by user u in test data
$\mathcal{U}^{te}$	A set of users in test data
$\mathcal{S}_u$	A sequence of items, $\mathcal{S}_u = \{i_u^1, i_u^2, \dots, i_u^{\lvert\mathcal{S}_u\rvert}\}$
i_u^t	The tth item in $\mathcal{S}_u$
$\hat{r}_{ui_u^t}$	Predicted preference of user u to item i_u^t
L	The order of Markov chains
ℓ	The ℓth order of Markov chains, $\ell \in \{1, 2, \dots, L\}$
$i_u^{t-\ell}$	The (t- ℓ)th item in $\mathcal{S}_u$
$\eta \in \mathbb{R}^{1\times L}$	Global weighting vector
$\eta^u \in \mathbb{R}^{1\times L}$	Personalized weighting vector w.r.t. user u
s_{ij}	Predefined similarity between item i and item j
λ	Trade-off parameter in mixed similarity
$d \in \mathbb{R}$	Number of latent dimensions
$V_{i\cdot}, W_{i\cdot} \in \mathbb{R}^{1\times d}$	Item-specific latent feature vector w.r.t. item i
$b_u \in \mathbb{R}$	User bias
$b_i \in \mathbb{R}$	Item bias
$\hat{r}_{ui}$	Predicted preference of user u to item i
γ	Learning rate
$\alpha_w, \alpha_v, \beta_\eta, \beta_v$	Trade-off parameters of regularization terms
T	Iteration number

(2) The *correlation* challenge. The learned similarity's weight of Fossil only considers the relative positions between the predicted item and its prior items, but neglects the correlations among them. Thus, one single type of learned similarity adopted in Fossil may not capture the correlations among the items well.

As a response to the challenges mentioned above, we combine a state-of-the-art model named Fossil with mixed similarity and design a novel and effective model called sequence-aware factored mixed similarity model (S-FMSM).

(1) For the *sparsity* challenge. Following Fossil, we adopt similarity models and high-order Markov chains to further utilize the item-item and user–item interactions. Because S-FMSM makes fuller use of the information, it can provide more accurate recommendations, which helps alleviate the sparsity issue.

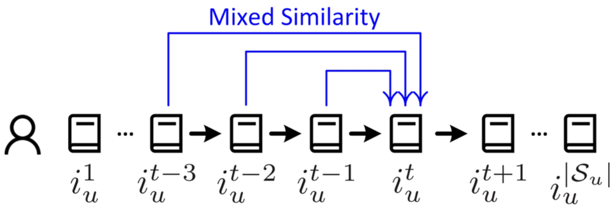

Fig. 3 Illustration of the relative positions in Fossil [3] and both the relative positions and the mixed similarity modeled in our S-FMSM. Notice that Fossil only utilizes the relative position correlations between the target item (i.e., i_u^t) and its preceding items, while our S-FMSM not only exploits the relative position correlations but also the similarity correlations as captured in the mixed similarity

(2) For the *correlation* challenge. Inspired by P-FMSM [13], we use mixed similarity to integrate items' predefined similarity and the items' learned similarity in a unified way.

4 Sequence-aware factored mixed similarity model (S-FMSM)

In this section, we present the proposed sequential recommendation model, named S-FMSM, which is a sequence-aware factored mixed similarity model to integrate an item-to-item cosine similarity into the factored Markov chains model [3].

The remarkable contribution of Fossil is to consider short-term sequential information via high-order Markov chains. The rationale behind the specific term $\eta_\ell + \eta_\ell^u$ is that each of the previous L locations should contribute with different weights to the high-order smoothness, lacking the weight contribution from the latest specific items. On the basis of Fossil, we introduce a mixed similarity to improve the recommendation effectiveness (see Fig. 3), which is defined as:

$$\bar{U}_{u\cdot}^{-i_u^t} = \frac{1}{\sqrt{|\mathcal{I}_u \backslash \{i_u^t\}|}} \sum_{i' \in \mathcal{I}_u \backslash \{i_u^t\}} W_{i'\cdot} + \sum_{\ell=1}^{L} (\eta_\ell + \eta_\ell^u)((1-\lambda) + \lambda s_{i_u^t i_u^{t-\ell}}) W_{i_u^{t-\ell}\cdot}, \tag{9}$$

where $(1-\lambda) + \lambda s_{i_u^t i_u^{t-\ell}}$ is our focus in this paper inspired by the mixed similarity [13]. It is a combination of a predefined similarity (i.e., cosine similarity) and a learned similarity (i.e., the inner production of two latent feature vectors). Here $s_{i_u^t i_u^{t-\ell}}$ is the cosine similarity between item i_u^t and item $i_u^{t-\ell}$. In fact, what it captures is the weight of the history item $i_u^{t-\ell}$ in contributing to the target item i_u^t. The trade-off parameter λ tuned among $\{0,0.2,0.4,0.6,0.8,1\}$ adjusts the influence of $s_{i_u^t i_u^{t-\ell}}$ in preference prediction. Notice that when $\lambda = 0$, it reduces to Fossil, and when $0 < \lambda \le 1$, the predefined similarity is capable of capturing the local relations among items.

From Fig. 3, we can see that (i) our S-FMSM makes fuller use of the information such as the item–item and user–item interactions, so as to provide more accurate recommendation, which helps alleviate the sparsity challenge; and (ii) by introducing the mixed similarity, our S-FMSM not only exploits the relative position correlations but also the similarity correlations, which effectively addresses the correlation challenge.

We keep the basic formula of Fossil and replace its user latent vector $\bar{U}_{u\cdot}^{-i_u^t}$ with Eq. (9) and derive the prediction rule of our S-FMSM:

$$\begin{aligned}\hat{r}_{ui_u^t} &= b_{i_u^t} + \bar{U}_{u\cdot}^{-i_u^t} V_{i_u^t\cdot}^T \\ &= b_{i_u^t} + \left[\frac{1}{\sqrt{|\mathcal{I}_u \backslash \{i_u^t\}|}} \sum_{i' \in \mathcal{I}_u \backslash \{i_u^t\}} W_{i'\cdot} \right. \\ &\quad \left. + \sum_{\ell=1}^{L} (\eta_\ell + \eta_\ell^u)((1-\lambda) + \lambda s_{i_u^t i_u^{t-\ell}}) W_{i_u^{t-\ell}\cdot} \right] V_{i_u^t\cdot}^T \end{aligned} \tag{10}$$

Notice that we firstly introduce a mixed similarity to a sequential recommendation method. Considering the universality of the mixed similarity, we may combine it with other sequential recommendation methods such as those based on matrix factorization or even deep learning techniques.

On the basis of the prediction rule of S-FMSM in Eq. (10), we design two variants of S-FMSM, i.e., S-FMSM with pairwise preference learning (S-FMSM(pai) for short) and S-FMSM with pointwise preference learning (S-FMSM(poi) for short), and present them in detail in the two following sections.

5 S-FMSM with pairwise preference learning

5.1 Objective function

Our primary goal is to rank the engaged items as high as possible. Since the pairwise preference over two items [23] is an explainable and natural assumption, we use the relationship $\hat{r}_{ui} > \hat{r}_{uj}$, which means that a user u is likely to prefer an item $i \in \mathcal{I}_u$ to an item $j \in \mathcal{I} \backslash \mathcal{I}_u$. We adopt a personalized pairwise ranking to keep the loss at a minimum. Then we reach the objective function,

$$\min_{\Theta} \sum_{u \in \mathcal{U}} \sum_{i_u^t \in \mathcal{S}_u, t \neq 1} \sum_{j \notin \mathcal{I}_u} f_{ui_u^t j}, \tag{11}$$

where $f_{ui_u^t j} = -\ln \sigma(\hat{r}_{ui_u^t} - \hat{r}_{uj}) + \frac{\alpha_v}{2}\left\|V_{i_u^t\cdot}\right\|^2 + \frac{\alpha_v}{2}\left\|V_{j\cdot}\right\|^2 + \frac{\alpha_w}{2}\sum_{i' \in \mathcal{I}_u}\|W_{i'\cdot}\|^2 + \frac{\alpha_w}{2}\sum_{\ell=1}^{L}\left\|W_{i_u^{t-\ell}\cdot}\right\|^2 + \frac{\beta_v}{2}b_{i_u^t}^2 + \frac{\beta_v}{2}b_j^2 + \frac{\beta_\eta}{2}\|\eta_\ell\|^2 + \frac{\beta_\eta}{2}\left\|\eta_\ell^u\right\|^2$ is a tentative objective function for a randomly sampled triple (u, i_u^t, j) via "first (u, i_u^t) then j", and $\Theta = \{V_{i\cdot}, W_{i\cdot}, b_i, \eta_\ell, \eta_\ell^u, i = 1, 2, \ldots, m; u = 1, 2, \ldots, n; \ell = 1, 2, \ldots, L\}$ are the model parameters to be learned.

5.2 Gradients and update rules

The minimization of the objective function in Eq. (11) can be solved by the commonly used stochastic gradient descent (SGD) algorithm.

The gradient of each parameter $\theta \in \Theta$, i.e., $\nabla\theta = \frac{\partial(f_{ui_u^t j})}{\partial\theta}$, is computed as follows:

$$\nabla b_{i_u^t} = \beta_v b_{i_u^t} + (-1)\sigma(\hat{r}_{uj} - \hat{r}_{ui_u^t}), \tag{12}$$

$$\nabla b_j = \beta_v b_j + \sigma(\hat{r}_{uj} - \hat{r}_{ui_u^t}), \tag{13}$$

$$\begin{aligned} \nabla V_{i_u^t\cdot} = \alpha_v V_{i_u^t\cdot} + (-1)\sigma(\hat{r}_{uj} - \hat{r}_{ui_u^t}) \Bigg[\frac{1}{\sqrt{|\mathcal{I}_u \backslash \{i_u^t\}|}} \\ \sum_{i' \in \mathcal{I}_u \backslash \{i_u^t\}} W_{i'\cdot} + \sum_{\ell=1}^{L} (\eta_\ell + \eta_\ell^u)((1-\lambda) + \lambda s_{i_u^t i_u^{t-\ell}}) W_{i_u^{t-\ell}\cdot} \Bigg], \end{aligned} \tag{14}$$

$$\begin{aligned} \nabla V_{j\cdot} = \alpha_v V_{j\cdot} + \sigma(\hat{r}_{uj} - \hat{r}_{ui_u^t}) \Bigg[\frac{1}{\sqrt{|\mathcal{I}_u|}} \sum_{i' \in \mathcal{I}_u} W_{i'\cdot} \\ + \sum_{\ell=1}^{L} (\eta_\ell + \eta_\ell^u)((1-\lambda) + \lambda s_{j i_u^{t-\ell}}) W_{i_u^{t-\ell}\cdot} \Bigg], \end{aligned} \tag{15}$$

$$\begin{aligned} \nabla \eta_\ell = \beta_\eta \eta_\ell + (-1)\sigma(\hat{r}_{uj} - \hat{r}_{ui_u^t}) W_{i_u^{t-\ell}\cdot} \cdot \\ [V_{i_u^t\cdot}^T((1-\lambda) + \lambda s_{i_u^t i_u^{t-\ell}}) - V_{j\cdot}^T((1-\lambda) + \lambda s_{j i_u^{t-\ell}})], \ell = 1, \dots, L, \end{aligned} \tag{16}$$

$$\begin{aligned} \nabla \eta_\ell^u = \beta_\eta \eta_\ell^u + (-1)\sigma(\hat{r}_{uj} - \hat{r}_{ui_u^t}) W_{i_u^{t-\ell}\cdot} \cdot \\ [V_{i_u^t\cdot}^T((1-\lambda) + \lambda s_{i_u^t i_u^{t-\ell}}) - V_{j\cdot}^T((1-\lambda) + \lambda s_{j i_u^{t-\ell}})], \ell = 1, \dots, L, \end{aligned} \tag{17}$$

$$\begin{aligned} \nabla W_{i'\cdot} = \alpha_w W_{i'\cdot} + (-1)\sigma(\hat{r}_{uj} - \hat{r}_{ui_u^t}) \Bigg[\frac{1}{\sqrt{|\mathcal{I}_u \backslash \{i_u^t\}|}} \\ V_{i_u^t\cdot} - \frac{1}{\sqrt{|\mathcal{I}_u|}} V_{j\cdot} \Bigg], i' \in \mathcal{I}_u \backslash \{i_u^t, i_u^{t-1}, \dots, i_u^{t-L}\}, \end{aligned} \tag{18}$$

$$\nabla W_{i_u^t\cdot} = \alpha_w W_{i_u^t\cdot} + (-1)\sigma(\hat{r}_{uj} - \hat{r}_{ui_u^t}) \left[-\frac{1}{\sqrt{|\mathcal{I}_u|}} V_{j\cdot} \right], \tag{19}$$

$$\begin{aligned} \nabla W_{i_u^{t-\ell}\cdot} = \alpha_w W_{i_u^{t-\ell}\cdot} + (-1)\sigma(\hat{r}_{uj} - \hat{r}_{ui_u^t})[(V_{i_u^t\cdot}((1-\lambda) + \lambda s_{i_u^t i_u^{t-\ell}}) \\ - V_{j\cdot}((1-\lambda) + \lambda s_{j i_u^{t-\ell}}))(\eta_\ell + \eta_\ell^u)], \ell = 1 \dots, L. \end{aligned} \tag{20}$$

For each sampled triple (u, i_u^t, j), we have the update rule for each parameter, i.e., $\theta = \theta - \gamma \nabla \theta$, where $\gamma > 0$ is the learning rate.

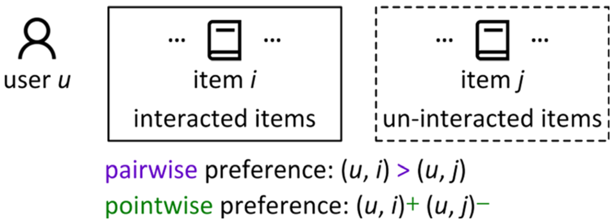

Fig. 4 The comparison of pairwise preference and pointwise preference. In pairwise preference assumption, we assume a user likes an interacted item more than an un-interacted one; and in pointwise preference assumption, we assume a user likes an interacted item and dislikes an un-interacted one

5.3 The algorithm of S-FMSM(pai)

The often adopted SGD algorithm utilized for solving the objective function in Eq. (11) is shown in Algorithm 1. We first initialize the model parameters (line 1) and calculate the similarity among the items (line 2). We have an outer loop ranges from 1 to T (line 3), and $|\mathcal{P}|$ inner iterations (line 4). In each inner iteration, we sample a (u, i_u^t) pair (line 4) and a negative item $j \in \mathcal{I}\backslash\mathcal{I}_u$ (line 5), calculate the gradients (line 6), and then update the corresponding model parameters (line 7).

Algorithm 1 The algorithm of S-FMSM(pai).

1: Initialize the model parameters.
2: Calculate the similarity among the items.
3: **for** $t = 1, \cdots, T$ **do**
4:     **for** each $(u, i_u^t) \in \mathcal{P}$ in a random order **do**
5:         Randomly pick up an item j from $\mathcal{I}\backslash\mathcal{I}_u$.
6:         Calculate gradients according to Eqs.(12-20).
7:         Update the model parameters $b_{i_u^t\cdot}$, $b_{j\cdot}$, $V_{i_u^t\cdot}$, $V_{j\cdot}$, η_ℓ, η_ℓ^u, $W_{i_u^t\cdot}$, $W_{i'\cdot}$, $W_{i_u^{t-\ell}\cdot}$, $i' \in \mathcal{I}_u\backslash\{i_u^t, i_u^{t-1}, \ldots, i_u^{t-L}\}, \ell = 1, 2, \ldots, L$.
8:     **end for**
9: **end for**

6 S-FMSM with pointwise preference learning

6.1 Objective function

In recommender systems, to obtain the optimal parameters of models, pairwise preference learning and pointwise preference learning are generally adopted. As shown in Fig. 4, pairwise preference assumes that a user prefers an interacted item to an un-interacted item, while pointwise preference assumes that a user likes his/her interacted items and dislikes his/her un-interacted items. For instance, if Mike buys an apple and does not buy a pear, we assume that Mike prefers apple to pear based on the pairwise preference assumption, while Mike likes apple and dislikes pear according to the pointwise one.

We call the pointwise variant of S-FMSM as S-FMSM(poi). The prediction rule of S-FMSM(poi) is equal to that of S-FMSM(pai), but it adopts the cross-entropy loss function. The objective function of S-FMSM(poi) is shown as follows:

$$\min_{\Theta} \sum_{u \in \mathcal{U}} \sum_{i \in \{i_u^t\} \cup \mathcal{N}} \log(1 + \exp(-r_{ui}\hat{r}_{ui})) + \mathcal{R}(\Theta), \tag{21}$$

where $\Theta = \{V_{i\cdot}, W_{i\cdot}, b_i, \eta_\ell, \eta_\ell^u, i = 1, 2, \ldots, m; u = 1, 2, \ldots, n; \ell = 1, 2, \ldots, L\}$ are the model parameters to be learned, $\mathcal{N}$ is a sampled unobserved item set to address the problem of lack of negative feedback, $r_{ui} = 1$ if $i = i_u^t$ and $r_{ui} = -1$ if $i \in \mathcal{N}$, $\hat{r}_{ui}$ is the predicted rating of user u on item i, and $\mathcal{R}(\Theta) = \frac{\alpha_v}{2}\left|\left|V_{i_u^t\cdot}\right|\right|^2 + \frac{\alpha_v}{2}\left|\left|V_{j\cdot}\right|\right|^2 + \frac{\alpha_w}{2}\sum_{i' \in \mathcal{I}_u} \left|\left|W_{i'\cdot}\right|\right|^2 + \frac{\alpha_w}{2}\sum_{\ell=1}^{L}\left|\left|W_{i_u^{t-\ell}\cdot}\right|\right|^2 + \frac{\beta_v}{2}b_{i_u^t}^2 + \frac{\beta_v}{2}b_j^2 + \frac{\beta_\eta}{2}\left|\left|\eta_\ell\right|\right|^2 + \frac{\beta_\eta}{2}\left|\left|\eta_\ell^u\right|\right|^2$ is a regularization term used to avoid overfitting.

6.2 Gradients and update rules

We follow the SGD algorithm used in S-FMSM(pai) to solve the minimization of the objection function in Eq. (21).

For a positive feedback pair (u, i), $i = i_u^t$, we have the gradients:

$$\nabla b_i = \beta_v b_i - \frac{1}{1 + \exp(\hat{r}_{ui})}, \tag{22}$$

$$\begin{aligned} \nabla V_{i\cdot} = \alpha_v V_{i\cdot} - \frac{1}{1 + \exp(\hat{r}_{ui})} \Bigg[& \frac{1}{\sqrt{|\mathcal{I}_u \backslash \{i\}|}} \sum_{i' \in \mathcal{I}_u \backslash \{i\}} W_{i'\cdot} \\ & + \sum_{\ell=1}^{L} (\eta_\ell + \eta_\ell^u)((1 - \lambda) + \lambda s_{i_u^t i_u^{t-\ell}}) W_{i_u^{t-\ell}\cdot} \Bigg], \end{aligned} \tag{23}$$

$$\nabla \eta_\ell = \beta_\eta \eta_\ell - \frac{1}{1 + \exp(\hat{r}_{ui})} W_{i_u^{t-\ell}\cdot}[V_{i\cdot}^T((1 - \lambda) + \lambda s_{i_u^t i_u^{t-\ell}})], \ell = 1, ..., L, \tag{24}$$

$$\nabla \eta_\ell^u = \beta_\eta \eta_\ell^u - \frac{1}{1 + \exp(\hat{r}_{ui})} W_{i_u^{t-\ell}\cdot}[V_{i\cdot}^T((1 - \lambda) + \lambda s_{i_u^t i_u^{t-\ell}})], \ell = 1, ..., L, \tag{25}$$

$$\begin{aligned} \nabla W_{i'\cdot} = \alpha_w W_{i'\cdot} - \frac{1}{1 + \exp(\hat{r}_{ui})} \left[\frac{1}{\sqrt{|\mathcal{I}_u \backslash \{i\}|}} V_{i\cdot} \right], \\ i' \in \mathcal{I}_u \backslash \{i_u^t, i_u^{t-1}, ..., i_u^{t-L}\}, \end{aligned} \tag{26}$$

$$\nabla W_{i_u^{t-\ell}\cdot} = \alpha_w W_{i_u^{t-\ell}\cdot} - \frac{1}{1+\exp(\hat{r}_{ui})}\Bigg[V_{i\cdot}((1-\lambda)+\lambda s_{i_u^t i_u^{t-\ell}}) \\ (\eta_\ell + \eta_\ell^u) + \frac{V_{i\cdot}}{\sqrt{|\mathcal{I}_u \backslash \{i\}|}}\Bigg], \ell = 1, \dots, L. \tag{27}$$

For a negative feedback pair $(u, i), i \in \mathcal{N}$, we have the gradients:

$$\nabla b_i = \beta_v b_i + \frac{\exp(\hat{r}_{ui})}{1+\exp(\hat{r}_{ui})}, \tag{28}$$

$$\nabla V_{i\cdot} = \alpha_v V_{i\cdot} + \frac{\exp(\hat{r}_{ui})}{1+\exp(\hat{r}_{ui})}\Bigg[\frac{1}{\sqrt{|\mathcal{I}_u|}}\sum_{i' \in \mathcal{I}_u} W_{i'\cdot} \\ + \sum_{\ell=1}^{L}(\eta_\ell + \eta_\ell^u)((1-\lambda)+\lambda s_{i_u^t i_u^{t-\ell}})W_{i_u^{t-\ell}\cdot}\Bigg], \tag{29}$$

$$\nabla \eta_\ell = \beta_\eta \eta_\ell + \frac{\exp(\hat{r}_{ui})}{1+\exp(\hat{r}_{ui})} W_{i_u^{t-\ell}\cdot}[V_{i\cdot}^T((1-\lambda)+\lambda s_{i_u^t i_u^{t-\ell}})], \ell = 1, ..., L, \tag{30}$$

$$\nabla \eta_\ell^u = \beta_\eta \eta_\ell^u + \frac{\exp(\hat{r}_{ui})}{1+\exp(\hat{r}_{ui})} W_{i_u^{t-\ell}\cdot}[V_{i\cdot}^T((1-\lambda)+\lambda s_{i_u^t i_u^{t-\ell}})], \ell = 1, ..., L, \tag{31}$$

$$\nabla W_{i'\cdot} = \alpha_w W_{i'\cdot} + \frac{\exp(\hat{r}_{ui})}{1+\exp(\hat{r}_{ui})}\Bigg[\frac{1}{\sqrt{|\mathcal{I}_u|}} V_{i\cdot}\Bigg], i' \in \mathcal{I}_u \backslash \{i_u^{t-1}, ..., i_u^{t-L}\}, \tag{32}$$

$$\nabla W_{i_u^{t-\ell}\cdot} = \alpha_w W_{i_u^{t-\ell}\cdot} + \frac{\exp(\hat{r}_{ui})}{1+\exp(\hat{r}_{ui})}\Bigg[V_{i\cdot}((1-\lambda)+\lambda s_{i_u^t i_u^{t-\ell}}) \\ (\eta_\ell + \eta_\ell^u) + \frac{V_{i\cdot}}{\sqrt{|\mathcal{I}_u|}}\Bigg], \ell = 1, \dots, L. \tag{33}$$

For either the positive or the negative sampled pair (u, i), we adopt the same update rule, i.e.,$\theta = \theta - \gamma \nabla \theta$, for each parameter, where $\gamma > 0$ is the learning rate.

6.3 The algorithm of S-FMSM(poi)

The SGD-based learning algorithm is depicted in Algorithm 2. Similar to that of S-FMSM(pai) in Algorithm 1, we first initialize the model parameters (line 1) and calculate the similarity among the items (line 2) and then have an outer loop ranges from 1 to T (line 3), and $|\mathcal{P}|$ inner iterations (line 4). The main difference is that we sample a set of items that have not been interacted by the user before (line 5). After that, we calculate the gradients (lines 7-8) and update the corresponding model parameters (line 9) for each item in the item set.

Algorithm 2 The algorithm of S-FMSM(poi).

```
1: Initialize the model parameters.
2: Calculate the similarity among the items.
3: for t = 1, ··· , T do
4:     for each (u, i_u^t) ∈ P in a random order do
5:         Randomly sample an unobserved item set N from I\I_u, |N| = 3.
6:         for i ∈ {i_u^t} ∪ N do
7:             If i = i_u^t, calculate gradients according to Eqs.(22-27).
8:             If i ∈ N, calculate gradients according to Eqs.(28-33).
9:             Update the model parameters b_{i_u^t·}, b_{j·}, V_{i_u^t·}, V_{j·}, η_ℓ, η_ℓ^u, W_{i·}, W_{i'·}, W_{i_u^{t-ℓ}·}, i' ∈
               I_u\{i_u^t, i_u^{t-1}, ..., i_u^{t-L}}, ℓ = 1, 2, ..., L.
10:        end for
11:    end for
12: end for
```

7 Experiments

7.1 Datasets

We adopt six commonly used public datasets from two domains in the experiments, including the MovieLens data and the Amazon e-commerce data.

MovieLens[1]: The MovieLens data are collected by the GroupLens Research team, containing users' rating records from the MovieLens website [2]. The MovieLens 100K (ML100K) dataset has been released since April 1998 and records the ratings (from 1 to 5) on 1,682 movies given by 943 users from September 1997 to April 1998. MovieLens 1M (ML1M) has been released since February 2003 and records the ratings on 3,952 movies given by 6,040 users from April 2000 to February 2003. Both of them are benchmark datasets for studies of recommendation algorithms because of their proper density. The average length of users' movie sequences in ML1M is larger than that in ML100K.

Amazon[2]: The Amazon e-commerce data collected by McAuley et al. [6, 14] contains product reviews and metadata (e.g., descriptions, category information and prices) from Amazon and is categorized into several classes. We choose four commonly used data covering different industries, including Office Products (Office), Automotive (Auto), Video Games (Video), and Cell Phones & Accessories (Cell), which are sparse in comparison with the MovieLens datasets.

Since our model is designed for implicit feedback, we treat all the observed behaviors as implicit feedback and preprocess each dataset as follows: (i) we remove the records of the users who rate fewer than five times; (ii) we remove the records of the items that are rated fewer than five times; (iii) we sort all the records according to

1 https://grouplens.org/datasets/movielens/.

2 http://jmcauley.ucsd.edu/data/amazon/.

Table 2 Statistics of the processed data used in the experiments, including the number of users (User #), the number of items (Item #), the number of records (Record #), and the density (Density)

Dataset	User #	Item #	Record #	Density (%)
ML100K	943	1,349	99,287	7.80
ML1M	6,040	3,416	999,611	4.80
Office	16,243	5,526	97,327	0.11
Video	30,935	12,111	260,163	0.07
Auto	31,877	9,992	122,009	0.04
Cell	67,453	17,969	346,245	0.03

the timestamps and split each user's sequence into three parts, i.e., the item(s) at the last step for test, the item(s) at the penultimate step for validation, and the remaining items for training. The statistics of the processed datasets are shown in Table 2. All the processed datasets and code used in our empirical studies are released.[3]

7.2 Evaluation metrics

We evaluate the recommendation performance of the algorithms via some commonly used ranking-oriented metrics for the recommendation lists of items, i.e., Precision@k (Pre@k), Recall@k (Rec@k), F1@k, NDCG@k, and 1-call@k. Notice that k stands for the length of the recommendation lists, which is fixed as $k = 20$ in the experiments.

– Pre@k is the percentage of accurately predicted items in a recommendation list,

$$\text{Pre}@k = \frac{1}{|\mathcal{U}^{te}|} \sum_{u \in \mathcal{U}^{te}} \frac{\sum_{\ell=1}^{k} \delta(i(\ell) \in \mathcal{I}_u^{te})}{k}, \tag{34}$$

where $\delta(x)$ is an indicator function with $\delta(x)$= 1 if x is true and $\delta(x)$= 0 otherwise, $i(\ell)$ represents the item that is located at the position ℓ of the recommendation list $\mathcal{I}_u^{re}$, and $\sum_{\ell=1}^{k} \delta(i(\ell) \in \mathcal{I}_u^{te})$ means the number of items in $\mathcal{I}_u^{re} \cap \mathcal{I}_u^{te}$.

– Rec@k is the percentage of accurate prediction in the test data,

$$\text{Rec}@k = \frac{1}{|\mathcal{U}^{te}|} \sum_{u \in \mathcal{U}^{te}} \frac{\sum_{\ell=1}^{k} \delta(i(\ell) \in \mathcal{I}_u^{te})}{|\mathcal{I}_u^{te}|}, \tag{35}$$

where $\sum_{\ell=1}^{k} \delta(i(\ell) \in \mathcal{I}_u^{te})$ means how many recommended items in $\mathcal{I}_u^{re}$ are also in $\mathcal{I}_u^{te}$. In the top-k recommended item list, $\sum_{\ell=1}^{k} \delta(i(\ell) \in \mathcal{I}_u^{te})$ denotes the number of true positive cases (TP), and $|\mathcal{I}_u^{te}|$ (i.e., the number of preferred items of user u) is the sum of the number of true positive cases and the number of false negative cases (i.e., TP+FN). Averaging all the recall values of the users in the test set, we get the equation of recall in Eq. (35).

[3] http://csse.szu.edu.cn/staff/panwk/publications/S-FMSM/.

– F1@k combines Pre@k and Rec@k, which is defined as follows,

$$\text{F1}_u@k = 2 \times \frac{\text{Pre}_u@k \times \text{Rec}_u@k}{\text{Pre}_u@k + \text{Rec}_u@k}. \tag{36}$$

– NDCG@k is commonly used to measure the position-aware ranking quality of a recommendation list,

$$\text{NDCG}@k = \frac{1}{|\mathcal{U}^{te}|} \sum_{u \in \mathcal{U}^{te}} \text{NDCG}_u@k. \tag{37}$$

The NDCG score of a specific user u is defined as follows,

$$\text{NDCG}_u@k = \frac{1}{Z_u} \text{DCG}_u@k, \tag{38}$$

where $\text{DCG}_u@k = \sum_{\ell=1}^{k} \frac{2\delta(i(\ell) \in \mathcal{I}_u^{te}) - 1}{\log(\ell + 1)}$ and Z_u is the best $\text{DCG}_u@k$ score with the preferred items in $\mathcal{I}_u^{te}$ in the beginning of $\mathcal{I}_u^{re}$.

– 1-call@k means whether there is at least one preferred item in a recommendation list,

$$1\text{- call}@k = \frac{1}{|\mathcal{U}^{te}|} \sum_{u \in \mathcal{U}^{te}} \delta\Big(\sum_{\ell=1}^{k} \delta(i(\ell) \in \mathcal{I}_u^{te}) \geq 1\Big). \tag{39}$$

7.3 Baseline methods

We compare our S-FMSM against five closely related recommendation methods:

– PopRank It uses the bias of each item as the prediction score and generates the same recommendation list for all the users, which is thus lack of personalization.
– BPR [23]: For recommendation with implicit feedback, BPR is a very competitive MF method with pairwise preference assumption, which optimizes the difference between the users' preferences for a positive sample and a negative sample.
– TransRec [5]: TransRec is a translation-based method which embeds items into a "transition space" and models users as translation vectors. Here we adopt $\mathcal{L}_2$ distance as metric and the probability that a user u transitions from item i_u^{t-1} to its next item i_u^t is estimated by $\hat{r}_{ui_u^t} = b_{i_u^t} - d(V_{i_u^{t-1}\cdot} + U_{u\cdot} + R_\cdot, V_{i_u^t\cdot})$, where $||V_{i_u^{t-1}\cdot}|| \leq 1$, $||V_{i_u^t\cdot}|| \leq 1$, $d(V_{i_u^{t-1}\cdot} + U_{u\cdot} + R_\cdot, V_{i_u^t\cdot}) = ||V_{i_u^{t-1}\cdot} + U_{u\cdot} + R_\cdot - V_{i_u^t\cdot}||$ is the Euclidean distance, and $R_\cdot$ is the global translation vector shared by all users.
– FISM [9]: In the factored item similarity model, the users' representations are learned through an item similarity matrix. Formally, the prediction rule of FISM is $\hat{r}_{ui} = \bar{U}_{u\cdot}^{-i} V_{i\cdot}^T + b_u + b_i$, where $\bar{U}_{u\cdot}^{-i} = \frac{1}{\sqrt{|\mathcal{I}_u \backslash \{i\}|}} \sum_{i' \in \mathcal{I}_u \backslash \{i\}} W_{i'\cdot}$ and $W_{i'\cdot} V_{i\cdot}^T$ is the learnable similarity between item i' and item i.

Table 3 Comparison among a popularity-based method, five factorization-based methods, and our S-FMSM

Property	PopRank	BPR	TransRec	FISM	FPMC	Fossil	S-FMSM
Personalized	×	√	√	√	√	√	√
Sequence-aware	×	×	√	×	√	√	√
Mixed similarity	×	×	×	×	×	×	√
Pairwise ranking	×	√	√	×	√	√	√
Translation-based	×	×	√	×	×	×	×

- FPMC [24]: FPMC is a combination of an MF [19] method and a first-order Markov chain, which models the sequential information in a factorization way. The probability that user u transfers from the last item i_u^{t-1} to its next item i_u^t is estimated by $\hat{r}_{ui_u^t} = U_{u\cdot}V_{i_u^t\cdot}^T + P_{i_u^{t-1}\cdot}Q_{i_u^t\cdot}^T$, where $P_{i_u^{t-1}\cdot}$ and $Q_{i_u^t\cdot}$ are two item transfer representations for the precursor and the successor, respectively.
- Fossil [3]: Fossil is a combination of a factored similarity method and a high-order Markov chains. The prediction of user u to item i_u^t is computed as $\hat{r}_{ui_u^t} = \bar{U}_{u\cdot}^{-i_u^t}V_{i_u^t\cdot}^T + b_{i_u^t}$, where the user's representation is $\bar{U}_{u\cdot}^{-i_u^t} = \frac{1}{\sqrt{|\mathcal{I}_u\backslash\{i_u^t\}|}}\sum_{i'\in\mathcal{I}_u\backslash\{i_u^t\}} W_{i'\cdot} + \sum_{\ell=1}^{L}(\eta_\ell + \eta_\ell^u)W_{i_u^{t-\ell}\cdot}$. And $\sum_{\ell=1}^{L}(\eta_\ell + \eta_\ell^u)W_{i_u^{t-\ell}\cdot}$ is associated with the high-order weighted transfer representations.

Table 3 shows the comparison between the aforementioned baseline methods and our S-FMSM.

7.4 Parameter configurations

For fair comparison, we fix the number of dimensions $d = 20$, the learning rate $\gamma = 0.01$ for all the factorization-based models in the experiments. And we adopt the commonly used stochastic gradient descent algorithm to train all the factorization-based methods. For FISM, we randomly sample a set of negative items $\mathcal{A}$ with $|\mathcal{A}| = 3|\mathcal{P}|$ following [9]. For BPR, TransRec, FPMC, Fossil and our S-FMSM, we use the same sampling strategy, i.e., randomly selecting one negative sample each time, for fair comparison.

For other hyper-parameters, we choose the trade-off parameter of the regularization terms $\alpha_v = \alpha_w = \beta_v = \beta_\eta$ from $\{0.1, 0.01, 0.001\}$, the order L from $\{1, 2, 3\}$ and the iteration number T from $\{100, 500, 1000\}$ via the NDCG@20 performance on the validation data. Moreover, the similarity trade-off λ in our S-FMSM is chosen from $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$ on the validation data. To ensure the reliability of these parameters, for each validation data, we select the optimal parameters according to the averaged performance of NDCG@20 of three runs. With the optimal parameter values, the final results on the test data are also averaged values of three runs.

Table 4 The searched best values of the parameters in each method on six different datasets. Notice that we fix $d = 20$, $\gamma = 0.01$ and select $\alpha_v = \alpha_w = \beta_v = \beta_\eta$ from $\{0.1, 0.01, 0.001\}$, L from $\{1, 2, 3\}$ and T from $\{100, 500, 1000\}$ via the NDCG@20 performance on the validation data

Configuration	Dataset	Method	Chosen parameters
$d = 20, \gamma = 0.01$	ML100K	PopRank	
		BPR	$\alpha_u = \alpha_v = \beta_v = 0.01, T = 1000$
		TransRec	$\alpha_u = \alpha_v = \beta_u = 0.01, T = 500$
		FISM	$\alpha_w = \alpha_v = \beta_v = \beta_u = 0.001, T = 500$
		FPMC	$\alpha_u = \alpha_v = \alpha_p = \alpha_q = 0.01, T = 1000$
		Fossil	$\alpha_w = \alpha_v = \beta_\eta = \beta_v = 0.001, L = 3, T = 500$
		S-FMSM	$\alpha_w = \alpha_v = \beta_\eta = \beta_v = 0.001, L = 3, \lambda = 1, T = 1000$
	ML1M	PopRank	
		BPR	$\alpha_u = \alpha_v = \beta_v = 0.01, T = 500$
		TransRec	$\alpha_u = \alpha_v = \beta_u = 0.01, T = 500$
		FISM	$\alpha_w = \alpha_v = \beta_v = \beta_u = 0.001, T = 100$
		FPMC	$\alpha_u = \alpha_v = \alpha_p = \alpha_q = 0.01, T = 1000$
		Fossil	$\alpha_w = \alpha_v = \beta_\eta = \beta_v = 0.001, L = 3, T = 100$
		S-FMSM	$\alpha_w = \alpha_v = \beta_\eta = \beta_v = 0.001, L = 3, \lambda = 0.8, T = 1000$
	Office	PopRank	
		BPR	$\alpha_u = \alpha_v = \beta_v = 0.1, T = 1000$
		TransRec	$\alpha_u = \alpha_v = \beta_u = 0.1, T = 1000$
		FISM	$\alpha_w = \alpha_v = \beta_v = \beta_u = 0.01, T = 1000$
		FPMC	$\alpha_u = \alpha_v = \alpha_p = \alpha_q = 0.01, T = 1000$
		Fossil	$\alpha_w = \alpha_v = \beta_\eta = \beta_v = 0.01, L = 2, T = 1000$
		S-FMSM	$T = 500$
	Auto	PopRank	
		BPR	$\alpha_u = \alpha_v = \beta_v = 0.1, T = 1000$
		TransRec	$\alpha_u = \alpha_v = \beta_u = 0.1, T = 1000$
		FISM	$\alpha_w = \alpha_v = \beta_v = \beta_u = 0.01, T = 1000$
		FPMC	$\alpha_u = \alpha_v = \alpha_p = \alpha_q = 0.1, T = 1000$
		Fossil	$\alpha_w = \alpha_v = \beta_\eta = \beta_v = 0.1, L = 1, T = 1000$
		S-FMSM	$\alpha_w = \alpha_v = \beta_\eta = \beta_v = 0.1, L = 1, \lambda = 0.2, T = 1000$
	Video	PopRank	
		BPR	$\alpha_u = \alpha_v = \beta_v = 0.01, T = 1000$
		TransRec	$\alpha_u = \alpha_v = \beta_u = 0.1, T = 1000$
		FISM	$\alpha_w = \alpha_v = \beta_v = \beta_u = 0.01, T = 1000$
		FPMC	$\alpha_u = \alpha_v = \alpha_p = \alpha_q = 0.01, T = 1000$
		Fossil	$\alpha_w = \alpha_v = \beta_\eta = \beta_v = 0.01, L = 2, T = 1000$
		S-FMSM	$\alpha_w = \alpha_v = \beta_\eta = \beta_v = 0.001, L = 2, \lambda = 0.8, T = 500$

Table 4 (continued)

Configuration	Dataset	Method	Chosen parameters
	Cell	PopRank	
		BPR	$\alpha_u = \alpha_v = \beta_v = 0.1, T = 1000$
		TransRec	$\alpha_u = \alpha_v = \beta_u = 0.1, T = 1000$
		FISM	$\alpha_w = \alpha_v = \beta_v = \beta_u = 0.01, T = 500$
		FPMC	$\alpha_u = \alpha_v = \alpha_p = \alpha_q = 0.1, T = 1000$
		Fossil	$\alpha_w = \alpha_v = \beta_\eta = \beta_v = 0.01, L = 1, T = 500$
		S-FMSM	$\alpha_w = \alpha_v = \beta_\eta = \beta_v = 0.01, L = 1, \lambda = 0.8, T = 1000$

For reproducibility, we report the searched optimal values of the parameters w.r.t. the NDCG@20 performance on the validation data in Table 4.

7.5 Experimental results

Table 5 shows the recommendation performance of six baseline methods and our S-FMSM with pairwise preference learning on six real-world datasets.

7.5.1 Main results

- Our model achieves the best performance on all the datasets except on Video where it is comparable with Fossil and on ML100K where it is comparable with TransRec, which shows the effectiveness of our proposed mixed similarity model. In order to be more clearly visible, we extract the two evaluation metrics, i.e., NDCG@20 and Rec@20, and show them in Fig. 5. From the histograms, we can see the effectiveness of our model over the compared methods. From the comparison between the factorization-based models and our S-FMSM in Table 3, we can see that only our S-FMSM incorporates the mixed similarity into the short-term preference.
- PopRank is a basic method that ranks items according to their popularities, which provides poor results due to its non-personalization as expected.
- For the non-sequential recommendation models, the pairwise approach BPR is better than the pointwise one FISM, which shows the advantage of the pairwise preference assumption.
- The sequential recommendation algorithms (i.e., TransRec, FPMC, Fossil and our S-FMSM) do not treat the user-interacted items as a bag of items, but rather as a sequence of items, which makes the models more powerful in terms of recommendation accuracy. For the denser datasets (i.e., ML100K and ML1M), Fossil performs better than FPMC, which shows that the high-order Markov chains captures more information than the lower order one. Moreover, FISM depicts the transitive relations between items which MF does not, so that it contributes to a

Table 5 Recommendation performance of PopRank, Bayesian personalized ranking (BPR) [23], translation-based recommendation (TransRec) [5], factored item similarity model (FISM) [9], factorizing personalized Markov chains (FPMC) [24], fusing similarity models with Markov chains (Fossil) [3], and our sequence-aware factored mixed similarity model (S-FMSM) with pairwise preference learning on six real-world datasets

Dataset	Method	Pre@20	Rec@20	F1@20	NDCG@20	1-call@20
ML100K	PopRank	0.0153	0.1197	0.0247	0.0619	0.2174
	BPR	0.0282±0.0004	0.1974±0.0049	0.0452±0.0007	0.1032±0.0012	0.3132±0.0043
	TransRec	**0.0287**±0.0001	0.2259±0.0012	0.0469±0.0001	0.1155±0.0007	0.3528±0.0016
	FISM	0.0275±0.0005	0.1832±0.0070	0.0439±0.0009	0.0985±0.0025	0.2930±0.0080
	FPMC	0.0273±0.0003	0.2292±0.0071	0.0452±0.0006	0.1147±0.0020	0.3574±0.0046
	Fossil	0.0277±0.0004	0.2299±0.0004	0.0458±0.0006	0.1153±0.0020	0.3627±0.0018
	S-FMSM	0.0286±0.0005	**0.2322**±0.0057	**0.0471**±0.0008	**0.1220**±0.0017	**0.3669**±0.0038
ML1M	PopRank	0.0067	0.0678	0.0116	0.0313	0.1133
	BPR	0.0113±0.0000	0.1102±0.0006	0.0196±0.0001	0.0518±0.0004	0.1827±0.0011
	TransRec	0.0149±0.0002	0.1448±0.0016	0.0259±0.0004	0.0656±0.0008	0.2327±0.0017
	FISM	0.0100±0.0001	0.0999±0.0007	0.0174±0.0002	0.0451±0.0007	0.1648±0.0016
	FPMC	0.0176±0.0002	0.1703±0.0026	0.0306±0.0004	0.0821±0.0003	0.2614±0.0039
	Fossil	0.0192±0.0002	0.1870±0.0028	0.0334±0.0004	0.0879±0.0005	0.2865±0.0047
	S-FMSM	**0.0194**±0.0003	**0.1895**±0.0019	**0.0337**±0.0004	**0.0895**±0.0015	**0.2879**±0.0036
Office	PopRank	0.0003	0.0040	0.0005	0.0013	0.0051
	BPR	0.0022±0.0001	0.0337±0.0013	0.0040±0.0002	0.0148±0.0007	0.0423±0.0017
	TransRec	0.0020±0.0000	0.0305±0.0004	0.0036±0.0000	0.0145±0.0004	0.0377±0.0003
	FISM	0.0021±0.0001	0.0340±0.0004	0.0040±0.0000	0.0154±0.0003	0.0413±0.0004
	FPMC	0.0020±0.0001	0.0297±0.0011	0.0037±0.0001	0.0133±0.0001	0.0353±0.0010
	Fossil	0.0023±0.0000	0.0346±0.0007	0.0042±0.0001	0.0153±0.0004	0.0428±0.0006
	S-FMSM	**0.0025**±0.0001	**0.0398**±0.0026	**0.0047**±0.0002	**0.0175**±0.0011	**0.0479**±0.0025
Auto	PopRank	0.0023	0.0362	0.0043	0.0173	0.0457
	BPR	0.0030±0.0001	0.0448±0.0017	0.0056±0.0002	0.0196±0.0004	0.0576±0.0019
	TransRec	0.0026±0.0000	0.0375±0.0004	0.0048±0.0001	0.0165±0.0004	0.0491±0.0005
	FISM	0.0029±0.0000	0.0435±0.0009	0.0054±0.0001	0.0188±0.0005	0.0560±0.0006
	FPMC	0.0019±0.0000	0.0266±0.0003	0.0034±0.0001	0.0117±0.0002	0.0353±0.0007
	Fossil	0.0032±0.0000	0.0467±0.0007	0.0059±0.0001	0.0209±0.0002	0.0612±0.0008
	S-FMSM	**0.0033**±0.0000	**0.0474**±0.0004	**0.0060**±0.0000	**0.0211**±0.0002	**0.0622**±0.0004
Video	PopRank	0.0024	0.0371	0.0044	0.0151	0.0467
	BPR	0.0050±0.0000	0.0734±0.0006	0.0092±0.0001	0.0319±0.0003	0.0928±0.0008
	TransRec	0.0052±0.0001	0.0771±0.0015	0.0095±0.0002	0.0330±0.0008	0.0957±0.0019
	FISM	0.0046±0.0000	0.0675±0.0005	0.0083±0.0001	0.0291±0.0003	0.0852±0.0006
	FPMC	0.0055±0.0001	0.0850±0.0010	0.0101±0.0001	0.0376±0.0004	0.1026±0.0010
	Fossil	**0.0059**±0.0000	**0.0895**±0.0002	0.0108±0.0001	0.0388±0.0001	0.1088±0.0005
	S-FMSM	0.0059±0.0001	0.0890±0.0010	**0.0108**±0.0001	**0.0398**±0.0006	**0.1089**±0.0009
Cell	PopRank	0.0027	0.0381	0.0050	0.0157	0.0532
	BPR	0.0033±0.0000	0.0447±0.0005	0.0060±0.0001	0.0203±0.0009	0.0617±0.0007
	TransRec	0.0034±0.0001	0.0474±0.0007	0.0062±0.0001	0.0220±0.0004	0.0622±0.0008
	FISM	0.0034±0.0001	0.0480±0.0008	0.0063±0.0001	0.0198±0.0004	0.0644±0.0012
	FPMC	0.0031±0.0001	0.0435±0.0006	0.0057±0.0001	0.0201±0.0004	0.0575±0.0009
	Fossil	0.0035±0.0001	0.0496±0.0013	0.0064±0.0002	0.0229±0.0003	0.0638±0.0017
	S-FMSM	**0.0040**±0.0000	**0.0568**±0.0007	**0.0074**±0.0001	**0.0265**±0.0005	**0.0737**±0.0007

The best results are marked in bold

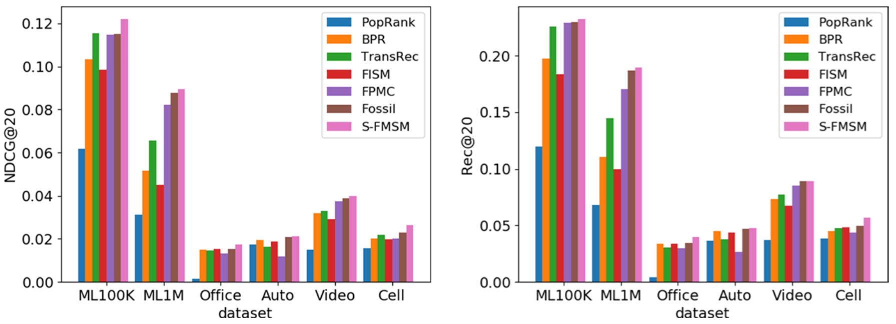

Fig. 5 Recommendation performance of our S-FMSM with pairwise preference learning and other methods on six real-world datasets (left: NDCG@20; right: Rec@20)

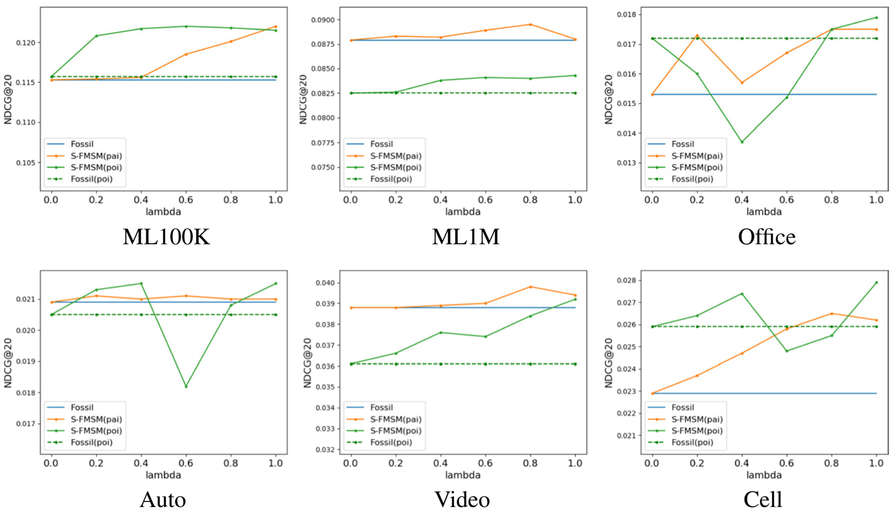

Fig. 6 Recommendation performance (NDCG@20) of our S-FMSM and Fossil with different values of $\lambda \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ on six real-world datasets

higher recommendation accuracy. In addition, the best value of the parameter L is small for the sparser datasets (i.e., Cell and Auto), which may be due to the fact that the sequential information between the items in a sparse data is not so strong.

- The results on the evaluation metrics are relatively small. This is because in the next-item recommendation task, only the item(s) at the last step of each user's interaction sequence is(are) put into the test set.
- Our model outperforms TransRec in all cases only except on the precision metric of ML100K, while Fossil is comparable with it in most cases. This showcases that our S-FMSM can provide more accurate recommendation than the different sequence modeling method.

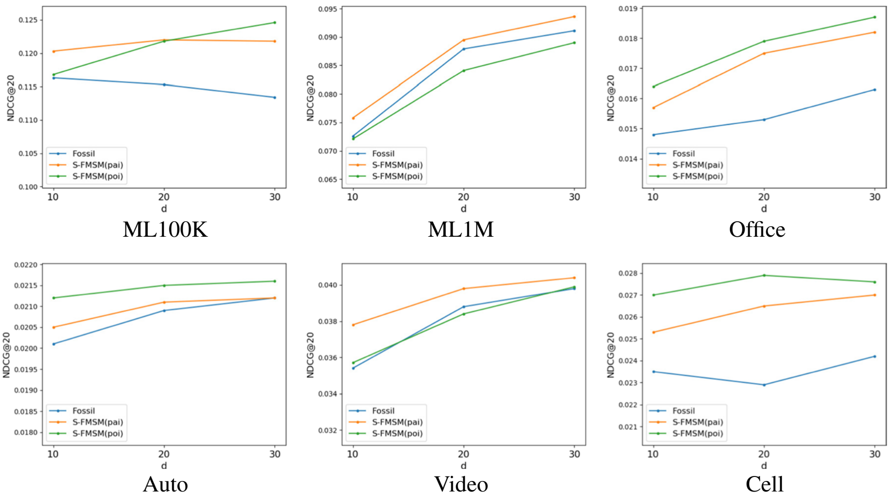

Fig. 7 Recommendation performance (NDCG@20) of our S-FMSM and Fossil with different numbers of latent dimensions $d \in \{10, 20, 30\}$ on six real-world datasets

- Fossil is a very strong baseline in the area of sequential recommendation. Our S-FMSM has achieved consistent improvement compared with Fossil on all the datasets, which again demonstrates the strength of our solution.

7.5.2 Effect of the parameter λ

In order to discern how mixed similarity shapes outcomes in the proposed model, we choose the trade-off parameters $\alpha_v = \alpha_w = \beta_v = \beta_\eta \in \{0.1, 0.01, 0.001\}$, $L \in \{1, 2, 3\}$, $T \in \{100, 500, 1000\}$ with a fixed λ from {0,0.2,0.4,0.6,0.8,1} via the NDCG@20 performance on the validation data. Then, we study our model with the optimal parameters on each test data for three times and report the averaged results.

As can be seen from Fig. 6, with different values of the parameter $\lambda \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$, the recommendation performance of our S-FMSM(pai) on the test data is better than that of Fossil in all cases. By adjusting the parameter λ of our S-FMSM(pai), we can see that when λ is 1 (ML100K), 0.8 (ML1M), 0.8 (Office), 0.2 (Auto), 0.8 (Video) and 0.8 (Cell), it achieves the best performance.

When $\lambda = 0$, our S-FMSM(poi) is reduced to Fossil(poi), and we mark Fossil(poi) as dotted line in Fig. 6. We can see that in most cases, S-FMSM(poi) with different λ outperforms Fossil(poi), in spite of some exceptional values. Nevertheless, S-FMSM(poi) always obtains more accurate recommendation performance than Fossil(poi) with a tuned value of λ.

Notice that λ is a parameter that needs to be tuned in a typical machine learning problem, which is usually data dependent. The experimental results also show that we need to choose the optimal value via a validation data, as we have done in the part of the main results.

Table 6 Recommendation performance of our S-FMSM(pai) and S-FMSM(poi) with different similarity measurements, including cosine similarity (cos) and Jaccard index (jac)

Dataset	Method	Pre@20	Rec@20	F1@20	NDCG@20	1-call@20
ML100K	cos, pai	0.0286±0.0005	0.2322±0.0057	0.0471±0.0008	0.1220±0.0017	0.3669±0.0038
	cos, poi	0.0290±0.0004	0.2359±0.0043	0.0477±0.0007	0.1218±0.0012	0.3680±0.0038
	jac, pai	0.0289±0.0002	0.2382±0.0009	0.0478±0.0003	0.1231±0.0008	0.3726±0.0024
	jac, poi	0.0286±0.0001	0.2323±0.0038	0.0472±0.0003	0.1206±0.0015	0.3644±0.0040
ML1M	cos, pai	0.0194±0.0003	0.1895±0.0019	0.0337±0.0004	0.0895±0.0015	0.2879±0.0036
	cos, poi	0.0186±0.0000	0.1803±0.0031	0.0324±0.0001	0.0841±0.0016	0.2766±0.0027
	jac, pai	0.0189±0.0004	0.1847±0.0028	0.0330±0.0007	0.0859±0.0021	0.2821±0.0047
	jac, poi	0.0186±0.0003	0.1827±0.0034	0.0324±0.0005	0.0841±0.0025	0.2790±0.0032
Office	cos, pai	0.0025±0.0001	0.0398±0.0026	0.0047±0.0002	0.0175±0.0011	0.0479±0.0025
	cos, poi	0.0026±0.0000	0.0404±0.0006	0.0048±0.0000	0.0179±0.0003	0.0495±0.0004
	jac, pai	0.0024±0.0001	0.0363±0.0008	0.0044±0.0001	0.0161±0.0002	0.0442±0.0009
	jac, poi	0.0023±0.0001	0.0354±0.0009	0.0043±0.0001	0.0149±0.0004	0.0425±0.0015
Auto	cos, pai	0.0033±0.0000	0.0474±0.0004	0.0060±0.0000	0.0211±0.0002	0.0622±0.0004
	cos, poi	0.0032±0.0000	0.0478±0.0002	0.0059±0.0000	0.0215±0.0003	0.0621±0.0002
	jac, pai	0.0033±0.0000	0.0472±0.0006	0.0060±0.0001	0.0210±0.0003	0.0619±0.0008
	jac, poi	0.0032±0.0000	0.0468±0.0001	0.0057±0.0000	0.0217±0.0002	0.0606±0.0002
Video	cos, pai	0.0059±0.0001	0.0890±0.0010	0.0108±0.0001	0.0398±0.0006	0.1089±0.0009
	cos, poi	0.0057±0.0000	0.0862±0.0009	0.0105±0.0001	0.0384±0.0005	0.1054±0.0010
	jac, pai	0.0059±0.0001	0.0889±0.0010	0.0107±0.0001	0.0385±0.0006	0.1082±0.0014
	jac, poi	0.0057±0.0000	0.0855±0.0003	0.0104±0.0000	0.0376±0.0001	0.1046±0.0003
Cell	cos, pai	0.0040±0.0000	0.0568±0.0007	0.0074±0.0001	0.0265±0.0005	0.0737±0.0007
	cos, poi	0.0042±0.0001	0.0586±0.0008	0.0077±0.0001	0.0279±0.0008	0.0767±0.0009
	jac, pai	0.0040±0.0001	0.0566±0.0006	0.0074±0.0001	0.0263±0.0003	0.0734±0.0010
	jac, poi	0.0040±0.0000	0.0561±0.0003	0.0073±0.0001	0.0255±0.0004	0.0733±0.0004

7.5.3 Impact of the number of latent dimensions *d*

In order to study the sensitivity of the model dimension d, we change the value of d from $\{10, 20, 30\}$ and choose the trade-off parameters via the NDCG@20 performance on the validation data. The results on test data are illustrated in Fig. 7.

From Fig. 7, we observe that our S-FMSM(pai) beats Fossil with varying latent dimensions on all datasets. For our S-FMSM(poi), it also beats Fossil on all datasets expect ML1M and Video; on Auto, Office, and Cell, it can even outperforms S-FMSM(pai). We can also find that a larger value of d usually leads to a better result, which is consistent with most previous studies on matrix factorized based methods. For a particular dataset, it is again a common practice to choose an appropriate value of d w.r.t. the performance on the validation data. Generally, it is believed that pairwise preference learning is superior to pointwise preference learning since it relaxes users' preference on items, while our experimental results show that which preference learning variant will have better performance is dependent on datasets. Therefore, we can decide which variant to use according to its performance on the validation data.

7.5.4 Impact of the similarity measurement

In order to study the effect of different similarity measurements, we replace cosine similarity with Jaccard index as predefined similarity in both two variants of S-FMSM and then obtain S-FMSM(jac, pai) and S-FMSM(jac, poi). After that, for two methods, we search optimal parameters $\alpha_v = \alpha_w = \beta_v = \beta_\eta \in \{0.1, 0.01, 0.001\}$, $L \in \{1, 2, 3\}, T \in \{100, 500, 1000\}$, and $\lambda \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ via the averaged NDCG@20 performance on the validation data of three runs. With optimal parameters, we run our models on test data and report the averaged results of three runs.

From Table 6, we can see that different similarity measurements lead to roughly similar recommendation performance, which indicates that our proposed similarity learning framework is robust with different similarity measurements.

8 Conclusions and future work

In this paper, we propose a novel sequence-aware recommendation method, i.e., sequence-aware factored mixed similarity model (S-FMSM), for a recently studied important task of next-item recommendation. We develop two variants, including a pairwise preference learning method S-FMSM(pai) and a pointwise preference learning method S-FMSM(poi). Our model considers a predefined similarity (e.g., cosine similarity or Jaccard index) between a recent item and the target item in order to better capture the short-term sequential effect. The main contribution is to study the mixed similarity in sequential recommendation. We mainly focus on factorization-based methods and conduct extensive empirical studies in the context of several state-of-the-art factorization-based methods on six real-world datasets and find that our S-FMSM achieves very promising performance.

For future works, we are interested in capturing more effect of users' short-term preferences by designing personalized variable-length subsequences. We are also interested in leveraging some auxiliary information such as social context [16, 30] so as to improve the recommendation performance for inactive users. Moreover, the concept of mixed similarity has a certain universality, and therefore, in the future, we plan to incorporate it into more sequential recommendation methods.

Acknowledgements We thank the handling editors and reviewers for their effort and constructive expert comments, and the support of National Natural Science Foundation of China Nos. 61872249 and 61836005.

References

1. Deshpande M, Karypis G (2004) Item-based top-N recommendation algorithms. ACM Trans. Inf. Syst. 22(1):143–177
2. Harper F, Konstan JA (2015) The movielens datasets: history and context. ACM Trans Interact Intell Syst 5(4):19:1–19:19

3. He R, McAuley J (2016) Fusing similarity models with Markov chains for sparse sequential recommendation. In: Proceedings of the 2016 IEEE 16th International Conference on Data Mining, ICDM ’16, pp 191–200
4. He R, Fang C, Wang Z, McAuley J (2016) Vista: a visually, socially, and temporally-aware model for artistic recommendation. In: Proceedings of the 10th ACM Conference on Recommender Systems, RecSys ’16, pp 309–316
5. He R, Kang W-C, McAuley J (2017) Translation-based recommendation. In: Proceedings of the 17th ACM Conference on Recommender Systems, RecSys ’17, pp 161–169
6. He R, McAuley J (2016) Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering. In: Proceedings of the 25th International Conference on World Wide Web, pp 507–517
7. He X, Liao L, Zhang H, Nie L, Hu X, Chua T-S (2017) Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp 173–182
8. Hidasi B, Karatzoglou A, Baltrunas L, Tikk D (2016) Session-based recommendations with recurrent neural networks. In: Proceedings of the 4th International Conference on Learning Representations, ICLR ’16
9. Kabbur S, Ning X, Karypis G (2013) 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, KDD ’13, pp 659–667
10. Koren Y (2008) Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 426–434
11. Li J, Ren P, Chen Z, Ren Z, Lian T, Ma J (2017) Neural attentive session-based recommendation. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM ’17, pp 1419–1428
12. Lin X, Chen H, Pei C, Sun F, Xiao X, Sun H, Zhang Y, Ou W, Jiang P (2019) A pareto-efficient algorithm for multiple objective optimization in e-commerce recommendation. In: Proceedings of the 13th ACM Conference on Recommender Systems, RecSys ’19, pp 20–28
13. Liu M, Pan W, Liu M ,Chen Y, Peng X, Ming Z (2017) Mixed similarity learning for recommendation with implicit feedback. Knowl-Based Syst 119(C):178–185
14. McAuley J, Targett C, Shi Q, Van Den Hengel A (2015) Image-based recommendations on styles and substitutes. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’15, pp 43–52
15. Pan R, Zhou Y, Cao B, Liu NN, Lukose RM, Scholz M, Yang Q (2008) One-class collaborative filtering. In: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15–19, 2008, Pisa, Italy, pp 502–511. IEEE Computer Society
16. Pan W, Chen L (2013) GBPR: group preference based Bayesian Personalized Ranking for one-class collaborative filtering. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI ’13, pp 2691–2697
17. Pan W, Ming Z (2017) Collaborative recommendation with multiclass preference context. IEEE Intell Syst 32(2):45–51
18. Pasricha R, McAuley J (2018) Translation-based factorization machines for sequential recommendation. In: Proceedings of the 12th ACM Conference on Recommender Systems, RecSys ’18, pp 63–71
19. Paterek A (2007) Improving regularized singular value decomposition for collaborative filtering. In: Proceedings of KDD Cup and Workshop, pp 39–42
20. Pazzani M, Billsus D (2007) Content-based recommendation systems. In: The Adaptive Web, pp 325–341. Springer
21. Quadrana M, Cremonesi P, Jannach D (2018) Sequence-aware recommender systems. ACM Comput Surv 51(4):66
22. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2009) BPR: Bayesian personalized ranking from implicit feedback. In: UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18–21, 2009, pp 452–461
23. Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L (2009) BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, UAI ’09, pp 452–461

24. Rendle S, Freudenthaler C, Schmidt-Thieme L (2010) Factorizing personalized Markov chains for next-basket recommendation. In: Proceedings of the 19th International Conference on World Wide Web, WWW '10, pp 811–820
25. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, WWW '01, pp 285–295
26. Song W, Xiao Z, Wang Y, Charlin L, Zhang M, Tang J (2019) Session-based social recommendation via dynamic graph attention networks. In: Proceedings of the 12th ACM International Conference on Web Search and Data Mining, WSDM '19, pp 555–563
27. Tang J, Wang K (2018) Personalized top-N sequential recommendation via convolutional sequence embedding. In: Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM '18, pp 565–573
28. Wang P, Guo J, Lan Y, Xu J, Wan S, Cheng X (2015) Learning hierarchical representation model for next basket recommendation. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15, pp 403–412
29. Ying H, Zhuang F, Zhang F, Liu Y, Xu G, Xie X, Xiong H, Wu J (2018) Sequential recommender system based on hierarchical attention networks. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI '18, pp 3926–3932
30. Zhao T, McAuley J, King I (2014) Leveraging social connections to improve personalized ranking for collaborative filtering. In: Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, pp 261–270
31. Zheleva E, Guiver J, Mendes REduarda, Milić-Frayling N (2010) Statistical models of music-listening sessions in social media. In: Proceedings of the 19th International Conference on World Wide Web, WWW '10, pp 1019–1028
32. Zhong L, Lin J, Pan W, Ming Z (2020) Proceedings of the sequence-aware factored mixed similarity model for next-item recommendation. In: Proceedings of the 2020 IEEE International Conference on Big Data and Smart Computing, pp 181–188
33. Zimdars A, Maxwell CD, Meek C (2001) Using temporal data for making recommendations. In: Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence, UAI '01, pp 580–588

Affiliations

Zhuoxin Zhan[1] · Liulan Zhong[1] · Jing Lin[1] · Weike Pan[1] · Zhong Ming[1]

✉ Weike Pan
panweike@szu.edu.cn

✉ Zhong Ming
mingz@szu.edu.cn

Zhuoxin Zhan
zhanzhuoxin2018@email.szu.edu.cn

Liulan Zhong
zhongliulan2017@email.szu.edu.cn

Jing Lin
linjing2018@email.szu.edu.cn

[1] National Engineering Laboratory for Big Data System Computing Technology and College of Computer Science and Software Engineering, Shenzhen University, 3688 Nanhai Avenue, Nanshan District, Shenzhen, People's Republic of China