TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 04/21 pp971–984
DOI: 10.26599/TST.2023.9010061
Volume 29, Number 4, August 2024

A Novel Popularity Extraction Method Applied in Session-Based Recommendation

Yuze Peng, Shengjun Xu, Qingkun Chen, Wenjin Huang, and Yihua Huang*

Abstract: Popularity plays a significant role in the recommendation system. Traditional popularity is only defined as a static ratio or metric (e.g., a ratio of users who have rated the item and the box office of a movie) regardless of the previous trends of this ratio or metric and the attribute diversity of items. To solve this problem and reach accurate popularity, we creatively propose to extract the popularity of an item according to the Proportional Integral Differential (PID) idea. Specifically, Integral (I) integrates a physical quantity over a time window, which agrees with the fact that determining the attributes of items also requires a long-term observation. The Differential (D) emphasizes an incremental change of a physical quantity over time, which coincidentally caters to a trend. Moreover, in the Session-Based Recommendation (SBR) community, many methods extract session interests without considering the impact of popularity on interest, leading to suboptimal recommendation results. To further improve recommendation performance, we propose a novel strategy that leverages popularity to enhance the session interest (popularity-aware interest). The proposed popularity by PID is further used to construct the popularity-aware interest, which consistently improves the recommendation performance of the main models in the SBR community. For STAMP, SRGNN, GCSAN, and TAGNN, on Yoochoose1/64, the metric P@20 is relatively improved by 0.93%, 1.84%, 2.02%, and 2.53%, respectively, and MRR@20 is relatively improved by 3.74%, 1.23%, 2.72%, and 3.48%, respectively. On Movieslen-1m, the relative improvements of P@20 are 7.41%, 15.52%, 8.20%, and 20.12%, respectively, and that of MRR@20 are 2.34%, 12.41%, 20.34%, and 19.21%, respectively.

Key words: popularity; Proportional Integral Differential (PID) algorithm; session-based recommendation; user's interests

1 Introduction

The concept of popularity has always been rooted in various types of recommendations. Relevant problems still keep hot research topics, such as popularity-based prediction[1] and popularity bias[2, 3]. Generally, popularity-related studies first define popularity and then turn to in-depth research. Therefore, the extraction of popularity can be treated as a foundation and is of great importance. In static data structure (time

• Yuze Peng, Shengjun Xu, Qingkun Chen, Wenjin Huang, and Yihua Huang are with School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China. E-mail: pengyz@mail2.sysu.edu.cn; xushj29@mail2.sysu.edu.cn; chenqk7@mail2.sysu.edu.cn; hwenjin@mail2.sysu.edu.cn; huangyih@mail.sysu.edu.cn.
• Yihua Huang is also with Guangdong Provincial Key Laboratory of Optoelectronic Information Processing Chips and Systems, Sun Yat-sen University, Guangzhou 510006, China, and Southern Marine Science and Engineering Guangdong Laboratory, Zhuhai 519080, China.
* To whom correspondence should be addressed.
Manuscript received: 2023-04-12; revised: 2023-05-26; accepted: 2023-06-12

information for interactions is not recorded), the popularity is usually defined as a numerical ratio or metric. Oh et al.[4] crawled the US gross box office of each movie from Internet Movie Database (IMDB) and used this gross box office as a metric to measure the popularity of a movie. Abdollahpouri et al.[5] regarded the popularity of an item as a simple ratio of users who have rated the item. Steck[6] defined the popularity of an item by the number of relevant (the user definitely likes these items) ratings obtained in the (unknown) complete data. These popularities are only the static feature and are usually combined with traditional recommendation methods (e.g., collaborative filtering and matrix factorization) to solve cold start and popularity bias, etc. However, for a given item, the meanings of the static ratio or metric with the different previous trends are different. For instance, if a ratio or metric is obtained from the previous upward trend and downward trend, it represents different popularity. Moreover, the same ratio or metric between items with different attributes also means different popularity. For example, although daily consumer goods and luxury goods have the same proportion of interactions, their popularity is different. Therefore, the static ratios and metrics cannot get insight into these attributes and trends, leading to a bias between the achieved and desirable popularity. Nonetheless, the static data without recording time information of interactions inhibit traditional popularity from considering the previous trends and the attributes, because clarifying these properties requires a long time span to be considered. However, with the diversification of recommended scenarios (e.g., Session-Based Recommendation (SBR) and sequential recommendation), corresponding data structures are temporal (time information for interactions is also recorded), which makes it possible to extract more accurate popularity. Therefore, to drive popularity extraction in a broader range, this paper resorts to the Proportional Integral Differential (PID)[7, 8] idea in the automation field to expand these aspects of popularity. In PID, the Integral (I) integrates a physical quantity over a time window, which agrees with the fact that determining the attributes of items also requires a long-term observation. The Differential (D) emphasizes an incremental change of a physical quantity over time, which coincidentally caters to a trend. Therefore, the obtained popularity is attribute- and trend-aware.

We further apply this attribute- and trend-aware popularity to SBR[9]. This is the first because SBR provides temporal data, which offers a potential possibility for extracting the trend- and attribute-aware popularity, to rely on. The second reason is that the user's interest is susceptible to popularity. Considering popularity when building interest is likely to further improve recommendation performance. Specifically, the session interest usually is divided into the general interest (global embedding) and the current interest (local embedding). The general interest is based on the whole item sequence of sessions and is usually long-term stable, resulting in it being almost impervious to popularity. However, the current interest is usually based on one or a few recent items, making it a temporary and dynamic feature and susceptible to sensitive factors such as popularity. To gain insight into the impact of popularity on the current interest, we carefully analyzed the sources of the current interest. The current interest comes from two aspects: (1) the personalized interest in the characteristics of the current item; (2) interest in the popularity of current items. Users currently interact with an item not only because they are interested in the item itself but also because of their response to its popularity. In particular, blockbuster-focused users[10] prefer popular items and are interested in some blockbusters. The niche-focused users[11] are actually interested in nonpopular items. Therefore, current interest is related to the characteristics and popularity of the current items. Nevertheless, none of the models consider popularity while constructing session interest to match the recommendation score in SBR.

As aforementioned, the extraction of popularity is critical in the recommendation community. It has been defined as a certain ratio or metric at the present moment without considering the previous trends and items' own attribute diversity. Absolute numerical values and time points are not flexible and sound for characterizing current popularity and tend to be oriented toward static data structures. In this paper, the idea of the PID is introduced for trend- and attribute-aware popularity extraction. Then, the proposed popularity is applied to SBR with an entry point of the user's interest, which generally improves the performance of major SBR models as a strategy. For clarity, the proposed architecture is given in Fig. 1. To summarize, the main contributions of the paper can be illustrated as follows:

- The PID idea is exploited to extract the attribute-

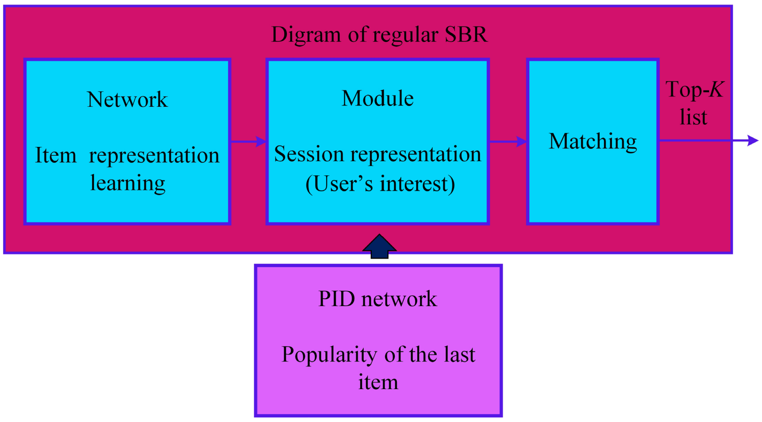


Fig. 1 Proposed architecture for SBR. On the basis of regular SBR architecture, the popularity element extracted by PID is creatively added to the user interest extraction module.

and trend-aware popularity. This novel idea drives popularity extraction in a broader range, providing a new idea for follow-up popularity research.

- A strategy of constructing popularity-aware session interest to match recommendation scores is proposed to improve SBR performance.
- Comprehensive experiments conducted on two real-world datasets consistently demonstrate the advantages of the proposed popularity over traditional popularity and the power of popularity-aware interest in the improvement of SBR.

2 Related Work

This section will review important works and concepts related to this research, including popularity studies and SBRs.

2.1 Popularity in recommendation

The concept of popularity has been widely used in early recommendations. It first appeared in the popularity-based recommendation[12] which is a type of recommendation around popularity calculation. The subject is to recommend hot items to users, effectively alleviating cold start[13] and other problems. In recent years, popularity is also used in news[14] and social recommendations[15]. Several existing online marketplaces also make use of popularity-based recommendations, for instance, Amazon, Yelp, TripAdvisor, SoundCloud, and Last.fm, and all suggest items that are currently trending.

In 2006, Ahn[16] exposed three key dimensions of popularity. The first dimension is the Average Rating (AR), which represents whether consumers perceive the product to be of high quality. It can be evaluated by dividing the sum of all buyer ratings of a product by the total number of buyers who have rated it. The second dimension is the Percentage of being Rated (PR), which represents the frequency of a product being interacted with regardless of its perceived value. It is strongly determined by seasons, festivals, trends, etc., and can be calculated by dividing the total number of ratings of a product by the total number of potential buyers. The third dimension is the size of the Strong-Support (SS) group for a product out of its average rating or frequency of interaction. It is the number of buyers who have shown strong support for a product divided by the total number of buyers who have rated it. A buyer whose rating for a product exceeds a predefined threshold value will join the strong-support group. In 2011, Steck[6] defined the popularity of an item by the ratio of relevant ratings (the highest rating) in the long-term static data record (complete data) when studying popularity bias. Also in 2011, Oh et al.[4] used the total earnings of a movie to reflect the popularity of a movie and further built a novel recommendation based on personal popularity tendency. In 2014, Kumar et al.[17] introduced contextual information to build a social popularity based SVD++ model with enhanced accuracy and scalability. In 2020, Abdollahpouri et al.[5] regarded the popularity of an item as a simple ratio of users who have rated the item when defining the different groups of users. In 2021, Zhang et al.[18] defined the global popularity of an item based on its interaction frequency and the local popularity as a ratio of interactions when researching popularity drift and bias. These approaches consider popularity as a numerical ratio or metric. Although different dimensions of popularity are highlighted (definition by Steck[6] emphasized AR, definition by Abdollahpouri et al.[5] emphasized PR, etc.), the dynamic trends of these ratios or metrics and items' own attributes are ignored. In addition, the essence is also to attribute popularity as a recommended criterion. However, this paper highlights the trends of these ratios and items' own attributes. In recommendation philosophy, popularity is regarded as a profile feature to enhance modeling session interests for SBR, which is distinguished from the popularity-based recommendation and opens up a novel scene.

2.2 SBR

The Recommendation System (RS) has appeared to deal with information overload issues, and some technologies in computer vision[19], biomedical[20], and social network sentiment analysis[21] have been introduced to improve RS. SBR is a type of recommendation system and predicts the next click or interaction based on anonymous sessions. Let $s=[v_{s,1},v_{s,2},...,v_{s,n}]$ be any anonymous session click sequence, where $v_{s,i}\in V$ is an item in the total item set V. A model M needs to be constructed so that the model has a corresponding output $\hat{y}=M(x)$ for any given prefix sequence $x=[v_{s,1},v_{s,2},...,v_{s,t}]$ $(1\leqslant t\leqslant n)$ of the session click sequence s. The output $\hat{y}=[y_1,y_2,...,y_{m-1},y_m]$ is a score or probability list over all the items in V, and y_i represents the recommendation probability of item $v_{s,i}$. The top-K $(1\leqslant K\leqslant m)$ high-scoring items in $\hat{y}$ are ranked in descending order to form a top-K recommendation list.

Since the emergence of deep learning, the mainline of SBR is extracting a hidden state (sequence pattern and session interest) from the historical sessions for later matching. The sequence patterns are mainly associated with Recurrent Neural Network (RNN) based methods, which take advantage of the natural advantages of RNN in sequence and include GRU4REC[22], NARM[23], and so on. The methods of extracting session interest include STAMP[24] and almost all Graph Neural Network (GNN) based models. Just in GNN-based methods, the general interest and the current interest are sometimes called global embedding and local embedding.

Recently, GNN-based models have become the major solutions for SBR. Wu et al.[25] posed the first work, SRGNN, employing a GNN modeling the complex transitions of items to optimize the embeddings of items. Finally, a soft attention mechanism was applied to the optimized embeddings to capture the general interest (global embedding). The current interest (local embedding) was represented by the embedding of the last item. Xu et al.[26] raised GCSAN, which further combined GNN with the self-attention mechanism to explore the effect of the historical interactions on the items. Therefore, GCSAN highlights the optimization of item embeddings. Afterward, Yu et al.[27] proposed TAGNN, which used target-aware attention to adaptively activate different user interests according to varied target items. Overall, the methods of extracting session interest models still focus on the front end to construct a more complex graph topology. However, when constructing the session interest, these major methods neither considered the other factors that are sensitive to the user's interest, so that the resulting session interest is sub-optimal.

3 Proposed Method

This section proposes the theoretical popularity model based on the PID idea. First, the traditional PID theory is introduced briefly, and the PID-based popularity extraction is analyzed in detail. Then, the mathematical model of popularity and the implementation under neural networks are described. Finally, the popularity-aware interest is proposed and further applied in SBR.

3.1 PID algorithm and popularity

PID[28] is one of the most important algorithms which are widely applied in the field of automatic control. The output has a discrete form[29] shown in Eq. (1):

$$u(k)=K_p\cdot \mathrm{err}(k)+K_i\cdot\sum \mathrm{err}(k)+K_d\cdot(\mathrm{err}(k)-\mathrm{err}(k-1)) \tag{1}$$

where K_p, K_i, and K_d denote adjustable parameters, and err(k) is the current error. The integral characterizes a long-term memory, while the differential is related to a variation trend.

After the emergence of deep learning, as datasets in most scenarios (SBR, click-through rate prediction, sequential recommendation, etc.) only provide an implicit interaction, the ratings are binarized so that the definitions of AR and SS are meaningless according to the defined calculation[16]. Thus, this paper only pays attention to PR, which is originally identified as a relative ratio, namely that the number of user-inspired behaviors is divided by the number of users. In this way, the popularity achieved by implicit interaction tends to be a current trend or prevalence (high exposure). However, although the popularity shrinks to PR, the current PR value is still not enough to reach advanced popularity. On one hand, due to the different attributes and prices of different types of items, the number of people to interact, with inherently, will vary greatly. This high exposure caused by attributes cannot represent the current trend. For example, given the same PR value between luxury handbags and common daily necessities such as toilet paper, the popularity should be distinct. Therefore, representing popularity only by the current PR is not fair and incurs a bias between the resulting and desirable popularity. More seriously, even though the PR of niche items increases sharply, they will be overwhelmed by popular daily products. On the other hand, items with the same PR values may keep the different trends in the past. The different trends in the past will lead to distinct popularity. Intuitively, in spite of the same PR, users are more likely to be attracted by those increasingly popular items, which means the popularity is still related to the distribution of past PR. Therefore, previous PR trends and item attributes are also expected to be responsible for the popularity of an item, while a static PR fails to consider these aspects.

This paper is inspired by the idea of PID, where I is the integration over a period of time and empowers a long-term memory property. Meanwhile, the attribute of an item can be intuitively reflected in the long-term interaction history. Therefore, a long-term PR can be tracked (compute the integral) to characterize the inherent attribute of an item. D is a differential, which is associated with an incremental property. It notes that the differential is simplified to the difference in the discrete-time system for simplicity. The long-term increment change implies a trend. Therefore, the concept of D is further extended to a long-term differential function curve (sequences of differences in discrete systems) to represent the previous trends. By doing this, the resulting curve and difference sequence are also memory signals. These memory properties help to catch precise and realistic popularity. Formally, the memory is reflected in that the popularity is not only related to current PR but past PR. Methodologically, the integral (sum) of PR in the past period is calculated for the attribute of items, and the differential function (difference sequence) is computed to imply a trend. A PR sequence value [0.01, 0.10, 0.20, 0.25, 0.95] and another [0.10, 0.25, 0.20, 0.01, 0.95] are given in the discrete situation to illustrate. Currently, they have the same PR value (0.95) (P). The PRs in the past period are summed, and the same sum 1.51 is obtained (I). The difference sequences are further calculated as the [0.09, 0.10, 0.05, 0.70] and [0.15, −0.05, −0.19, 0.94] (D). They are different, implying different trends. The attribute- and trend-aware component helps to accurately extract popularity. The theoretical model and the form under neural network will be given in the following chapter.

3.2 Mathematical model of the popularity

In this section, the general mathematical formalization of popularity extraction and the form under the neural networks are given.

Assume that the dynamic interactions $\tilde{x} = \tilde{x}(t)$ are a continuous function of time, and the cumulative number of interactions n of an item and time t satisfy $n = m(t)$. Obviously, $m(t)$ keeps an integral relationship with $\tilde{x}(t)$, that is, $m(t)=\int \tilde{x}(t)\,\mathrm{d}t$. To facilitate processing, $\tilde{x}(t)$ is discretized into a discrete time function by sampling and accumulating in different time periods, that is, the dynamic interactions have a discrete form $x[t_i] = m(t_i) - m(t_{i-1})$ or $x[t_i] = \int_{t_{i-1}}^{t_i} \tilde{x}(t)\,\mathrm{d}t$, $\Delta t = t_i - t_{i-1}$. At the same time, all online records (users or sessions) during the corresponding Δt period are counted to form a sequence or discrete function $p[t]$. The PR value at time t_i is evaluated as Eq. (2):

$$\mathrm{PR}[t_i] = \frac{x[t_i]}{p[t_i]} \tag{2}$$

Then, the differentiation is transformed into a difference signal for the discrete-time function. From the above analysis, the item popularity y can be derived from the discrete function $x[t]$ and sequence $p[t]$. The discretized PID form turns to Eq. (3):

$$y(t_i) = K_p \cdot \frac{x[t_i]}{p[t_i]} + K_i \cdot \sum_{n=N-1}^{i} \frac{x[t_n]}{p[t_n]} + K_d \cdot \left(\frac{x[t_i]}{p[t_i]} - \frac{x[t_{i-1}]}{p[t_{i-1}]}\right) \tag{3}$$

The current popularity $y(t_i)$ is related to the inherent value p (the PR value at the current moment $x[t_i]/p[t_i]$), the integral value i in the past period ($\sum_{n=N-1}^{i} x[t_n]/p[t_n]$), and the difference d ($x[t_i]/p[t_i] - x[t_{i-1}]/p[t_{i-1}]$). As mentioned in Section 3.1, the current difference is extended to a difference sequence to imply a long-term trend. Then, the popularity can be written as

$$y(t_i)=f\left(\frac{x[t_i]}{p[t_i]},\sum_{n=N-1}^{i}\frac{x[t_n]}{p[t_n]},\left[\frac{x[t_{i-(k-1)}]}{p[t_{i-(k-1)}]}-\frac{x[t_{i-k}]}{p[t_{i-k}]},\ldots,\frac{x[t_i]}{p[t_i]}-\frac{x[t_{i-1}]}{p[t_{i-1}]}\right]\right) \tag{4}$$

where N is the integration time limit, and k is the length of the difference sequence.

Under the framework of neural networks, a Gated Recurrent Unit (GRU) is employed to extract the feature of the difference sequence $[x[t_{i-(k-1)}]/p[t_{i-(k-1)}]-x[t_{i-k}]/p[t_{i-k}],\ldots,x[t_i]/p[t_i]-x[t_{i-1}]/p[t_{i-1}]]$. The output in the last step represents the long-term trend. The inducing result is shown as

$$\Delta\boldsymbol{x}=\mathrm{GRU}\left(\frac{x[t_{i-(k-1)}]}{x[t_{i-(k-1)}]}-\frac{x[t_{i-k}]}{x[t_{i-k}]},\ldots,\frac{x[t_i]}{x[t_i]}-\frac{x[t_{i-1}]}{x[t_{i-1}]}\right) \tag{5}$$

Thus, the function in Eq. (4) can be generalized to Eq. (6):

$$c=K_p\cdot p+K_i\cdot i+K_d\cdot\Delta\boldsymbol{x} \tag{6}$$

where c denotes the current popularity. p is $x[t_i]/p[t_i]$, i is $\sum_{n=N-1}^{i}x[t_n]/p[t_n]$, and $\Delta\boldsymbol{x}$ is the hidden state of GRU at the last step. The final form under neural networks of Eq. (6) can be written as

$$\boldsymbol{c}=\boldsymbol{W}[\boldsymbol{P}\|\boldsymbol{I}\|\Delta\boldsymbol{x}]+\boldsymbol{b} \tag{7}$$

where $\boldsymbol{P}$ and $\boldsymbol{I}$ are the vector augmentations of p and i, respectively. Note that vector augmentation in this article refers to copying the numerical value to the numerical value to each position of the vector. $[\cdot\|\cdot]$ represents concatenation operation. K_p, K_i, and K_d are parameterized into learnable weight $\boldsymbol{W}$ and bias $\boldsymbol{b}$ in neural networks and are further updated via back-propagation of the gradient. The workflow of popularity extraction under neural networks is shown in Fig. 2.

3.3 Applying the popularity to SBRs

This section will reconstruct a popularity-aware interest to use for SBR. Generally, the user's interest consists of the current interest and the general interest.

Current interest. As aforementioned, the current interest comes from the personalized interest in the features of current items and the response to the popularity of current items. Existing approaches only emphasize personalized interest and use the embedding of the last-visited item to represent the current interests i.e., $\boldsymbol{s}_l=\boldsymbol{v}_{s,n}$[23–25, 27]. But they ignore that in addition to users' own personalized interests, users' interests can also come from the popularity of items. Therefore, the impact of popularity on current interest is additionally considered in this paper. Specifically, the popularity of the last-visited item is extracted according to the theory in Section 3.2. In PID implementation, the appropriate is chosen to get the discrete functions $x[t]$ and $p[t]$. Then, integration time limit N and the length of difference sequence k are determined. After that, the inherent value p, the integral value i, and the hidden state $\Delta\boldsymbol{x}$ are calculated as required based on the theory in Section 3.2. Finally, the popularity $\boldsymbol{c}$ is obtained according to Eq. (7).

Specifically, the current interest $\boldsymbol{s}_l$ is achieved by taking linear transformation over the concatenation of the embedding $\boldsymbol{v}_{s,n}$ and the popularity $\boldsymbol{c}$ vectors, as shown in Eq. (8):

$$\boldsymbol{s}_l=\boldsymbol{W}[\boldsymbol{v}_{s,n}\|\boldsymbol{c}] \tag{8}$$

Note that this paper only proposes a popularity-aware interest strategy. For a few sophisticated models with current interests involving a few recent items, averaging the popularity of these items probably can be roughly considered in spite of no experimental proof.

General interest. A soft-attention mechanism is used to aggregate all involved node vectors with different levels of priority. As shown in Eq. (9):

$$\alpha_i=\boldsymbol{q}^{\mathrm{T}}\sigma(\boldsymbol{W}_1\boldsymbol{v}_n+\boldsymbol{W}_2\boldsymbol{v}_i+\boldsymbol{b}),\quad \boldsymbol{s}_g=\sum_{i=1}^{n}\alpha_i\boldsymbol{v}_i \tag{9}$$

where $\boldsymbol{v}_i$ is the embedding of item i in session, and $\boldsymbol{v}_n$

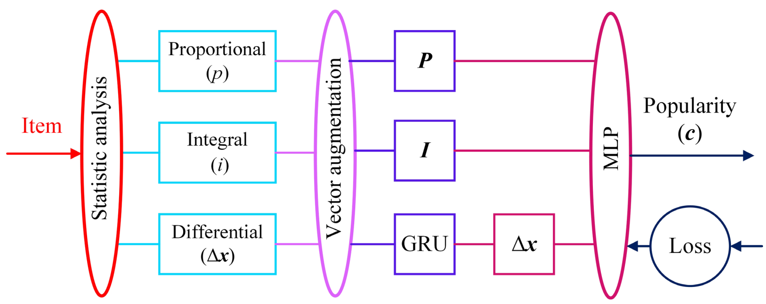


Fig. 2 Neural network framework of PID for popularity extraction.

is the embedding of the last item n in session. s_g is the general interest, α_i is the attention score, and σ is the sigmoid function. The parameters $\boldsymbol{q}, \boldsymbol{b} \in \mathbb{R}^d$ and $\boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}^{d \times d}$ control the weights of item embedding vectors. Because the general interest is relatively stable and not easily influenced by popularity, we do not consider the impact of popularity when modeling general interest.

Finally, the hybrid embedding (session interest) can be represented as

$$\boldsymbol{s}_h = \boldsymbol{W}_3[\boldsymbol{s}_g \| \boldsymbol{s}_l] \tag{10}$$

where $\boldsymbol{W}_3 \in \mathbb{R}^{d \times 2d}$ controls the linear transformation, and $\mathbf{s}_h$ is the popularity-aware interest.

4 Experiment Setup

To verify the effectiveness of the proposed method and strategy, comprehensive experiments are conducted and aim to answer the following questions:

(1) Does the proposed popularity-aware interest really work and improve recommendation performance?

(2) Does trend- and attribute-aware popularity have advantages over traditional popularity?

(3) How does popularity-aware interest affect the final recommendation result?

Methodologically, bare interest (without considering popularity), interest enhanced by traditional popularity, and interest enhanced by attribute- and trend-aware popularity are used to match recommendation scores for the next recommendation. The matching process follows the general inner product and can be defined as

$$\hat{z}_t = \boldsymbol{s}_h^{\mathrm{T}} \boldsymbol{v}_t \tag{11}$$

where $\boldsymbol{s}_h$ represents one of the above session interests, and $\hat{z}_t$ is the recommendation score of candidate item v_t. The recommendation score for all candidates forms the scores list z. Then a softmax function is used to transform z into a recommendation probability distribution $\hat{\boldsymbol{y}}$:

$$\hat{\boldsymbol{y}} = \mathrm{softmax}(z) \tag{12}$$

Top-K probabilities in $\hat{\boldsymbol{y}}$ will be ranked in descending order to form the final recommended list.

For each session, the cross-entropy of the prediction and the ground truth serves as the loss function and can be expressed as

$$L(\hat{\boldsymbol{y}}) = -\sum_{i=1}^{m} \boldsymbol{y}_i \log(\hat{\boldsymbol{y}}_i) + (1-\boldsymbol{y}_i)\log(1-\boldsymbol{y}_i) \tag{13}$$

where $\boldsymbol{y}$ denotes the one-hot encoding vector of the ground truth item. The Backpropagation Through Time (BPTT) algorithm is adopted to train the model. The three questions above will be answered based on corresponding recommendation results.

4.1 Dataset

Two real-world representative datasets are employed to evaluate the proposed method, i.e., Yoochoose※ and Movieslen-1m§. The Yoochoose dataset is obtained from the RecSys Challenge 2015 and contains a stream of user clicks on an e-commerce website within six months. The Movieslen-1m is also a widely-used benchmark for evaluating recommendation algorithms and contains rating data of multiple users on multiple movies. This paper treats the presence of a review or rating as implicit feedback and determines the sequence order of actions according to timestamps. For a fair comparison, the data preprocessing scheme in Refs. [24, 25, 27] is closely followed. Specifically, both items with fewer than five occurrences and sessions consisting of less than 2 items are dropped. For generating training and testing sets, sessions of the last days are used as the testing set for Yoochoose, and sessions of the last months as the testing set for Movieslen-1m. Furthermore, clicks (items) that did not appear in the training set are filtered out. Then, the sequence splitting technology is applied to produce sequences (sessions) and corresponding labels. Following Refs. [24, 25, 27], we only report the results trained on more recent fractions 1/64 of training sequences. It notes that the average session length is relatively large in Movieslen-1m, as a result, we only keep the most recent 50 actions once the length exceeds 50. The statistics of the datasets are summarized in Table 1.

4.2 Benchmark

These baselines are representative models in SBR and the state-of-the-art models in the past. They are based on complex neural algorithms (e.g., RNN, GNN, and attention mechanism) and are powerful enough in representation learning.

• **STAMP**[24] is capable of capturing users' general interests from the long-term memory of a session context and taking users' current interests into account from the short-term memory of the last click. It is

※http://2015.recsyschallenge.com/challege.html
§https://grouplens.org/datasets/movielen

Table 1 Statistics of two datasets used in the experiments.

Dataset	Number of clicks	Number of training sessions	Number of test sessions	Number of items
Yoochoose1/64	557 248	369 859	55 898	16 766
Movieslen-1m	–	913 351	80 220	3416

completely based on the attention mechanism.

• **SRGNN**[25] processes the item sequence in a session as a session-induced graph. In particular, the general interest and the current interest are specifically called global embedding and local embedding. Similarly, an attention mechanism is applied to the whole item sequence to extract general interest. The embedding of the last click represents the current interest.

• **GCSAN**[26] indicates that the self-attention mechanism with graph neural network can adaptively assign weights to focus on long-range dependencies or more recent actions, while the short-term interest is also indispensable for improving the recommendation performance.

• **TAGNN**[27] is the only attempt of applying target-aware attention to learn interest representation varied with different target items. It holds a similar route when building interests for matching recommendation scores.

These main models construct session interests based on the characteristics of sequence items, regardless of other external factors that are sensitive to the user's interests.

4.3 Evaluation metrics

The final recommendation performance is characterized by P@K and MRR@K. They are two widely-used metrics in recommendation scenarios.

• **P@*K*** is the recall proportion of sessions whose predicted top-K list includes the target item on the whole testing set. It can be evaluated as

$$\mathrm{P}@K = \frac{1}{\mathrm{Testing}} \sum_{i=1}^{|\mathrm{Testing}|} \mathrm{Hit}_i@K \tag{14}$$

If the top-K list includes the target item, $\mathrm{Hit}_i@K = 1$, otherwise, $\mathrm{Hit}_i@K = 0$. |Testing| is the size of the testing set.

• **MRR@*K*** emphasizes the ranking of the target items in the top-K lists. It can be evaluated as

$$\mathrm{MRR}@K = \frac{1}{\mathrm{Testing}} \sum_{i=1}^{|\mathrm{Testing}|} \frac{1}{\mathrm{rank}_i@K} \tag{15}$$

where $\mathrm{rank}_i@K$ represents the ranking of the target item in the predicted top-K list. If the ranking goes beyond the top-K list, $\frac{1}{\mathrm{rank}_i@K} = 0$. Therefore, MRR@$K$ is a more precise and stricter metric.

By default, we also set $K = 20$ to evaluate and further compare the model performance.

4.4 Hyperparameter setting

Following previous methods[24, 25, 27], this paper sets the dimensionality of latent vectors d=100 for experiments. The mini-batch for mini-training is 30. All parameters are optimized by an Adam optimizer, in which the initial learning rate is set to 0.001 and will decay by 0.1 after every 3 epochs. Moreover, the L_2 penalty is set to 105. All experiments are conducted on the platform of a single GTX-1080 Ti GPU.

5 Result and Analysis

5.1 Effectiveness of popularity-aware interest

To verify the effectiveness of the popularity-aware interest, we used popularity-aware interest to replace bare interest to match the recommendation score. The performance comparison across all baselines is shown in Table 2. The metrics of models with popularity-aware interest are in parentheses, and the relative improvement in performance is shown in bold. All other conditions are the same and optimal.

According to the observations in Table 2, the following conclusions can be drawn:

Table 2 Overall performance comparison over two datasets (p < 0.02).

Model	Yoochoose1/64		Movieslen-1m	
	p@20 (%)	MRR@20 (%)	p@20 (%)	MRR@20 (%)
STAMP	68.47 (69.38) **0.93↑**	29.67 (30.78) **3.74↑**	15.51 (16.66) **7.41↑**	5.98 (6.12) **2.34↑**
SRGNN	70.57 (71.87) **1.84↑**	30.94 (31.32) **1.23↑**	23.00 (26.57) **15.52↑**	7.01 (7.88) **12.41↑**
GCSAN	70.47 (71.89) **2.02↑**	30.18 (31.00) **2.72↑**	24.03 (26.00) **8.20↑**	6.98 (8.40) **20.34↑**
TAGNN	70.85 (72.64) **2.53↑**	30.78 (31.85) **3.48↑**	24.11 (28.96) **20.12↑**	7.60 (9.06) **19.21↑**

Note: The metrics of models with popularity-aware interest are in parentheses, and the relative improvement in performance is shown in bold. All other conditions are the same and optimal.

- For STAMP, SRGNN, GCSAN, and TAGNN, on Yoochoose1/64, the metric P@20 is relatively improved by 0.93%, 1.84%, 2.02%, and 2.53%, and MRR@20 is relatively improved by 3.74%, 1.23%, 2.72%, and 3.48%, respectively. On Movieslen-1m, the relative improvement of P@20 is 7.41%, 15.52%, 8.20%, and 20.12%, and that of MRR@20 is 2.34%, 12.41%, 20.34%, and 19.21%, respectively. Firstly, the seemingly small improvement in recommended performance can bring huge benefits to merchants, as the number of consumers is enormous. Secondly, the performance improvement of previous models is also slight, such as TAGNN compared to SRGNN. Finally, to demonstrate the reliability of this improvement, rather than being caused by random factors, we conduct a paired student t-test on the results. Specifically, each group of experiments is randomly initialized and run 5 times to obtain 5 results. All the resulting paired student t-tests results are p<0.02, indicating that this performance improvement is significantly reliable and not caused by accidental factors. In the end, it can be seen that using popularity-aware interest to recommend has been steadily rewarded with superior recommendation results on Yoochoose1/64 and Movieslen-1m. The consistent reward indicates that popularity is a sensitive factor for the user's interest and that the proposed popularity-aware interest strategy is effective and generic. Especially in scenes like movie watching where popularity is important, popularity-aware interest can greatly improve performance.
- The advanced recommendation performance is achieved by constructing complex graph structures (e.g., SRGNN better than STAMP, and GCSAN better than SRGNN), which is because of the increased expressiveness of the network on representation learning. But the proposed popularity-aware interest can further improve their performance consistently within a range in a beyond sequence pattern and graph topology manner.

Therefore, the first question has been answered. Popularity-aware interest is an effective strategy for improving the final recommendation results.

5.2 Advantage of the trend- and attribute-aware popularity over traditional popularity

This section will investigate that the trend- and attribute-aware popularity extracted by PID has advantages over traditional popularity. To this end, we select the TAGNN as the infrastructure and further regulate and control the Δt, integration time limit N, as well as the number of the difference terms k. N and k are associated with the attribute-aware and trend-aware characteristics, respectively. To facilitate horizontal and vertical comparisons, we set Δt to 32 days, 8 days, 2 days, 0.5 days, and 0.125 days, respectively, N to 4, and k to 3. For vertical comparison, for each Δt, N and k are added to compare with the conditions of $N=0$ and $k=0$. For horizontal comparison, the value of Δt just satisfies the N-fold relation so that the traditional popularity is compared with the popularity obtained by the PID idea within a fixed period of time. For example, conditions of Δt = 32 days, $N=0$, and $k=0$ mean that the traditional popularity will be obtained by computing the PR value within the last 32 days. Also in the last 32 days, conditions of Δt = 8 days, $N=4$, $k=3$ mean the attribute- and trend-aware popularity will be achieved. The resulting results on Movieslen-1m and Yoochoose1/64 dataset are shown in Figs. 3 and 4.

In Figs. 3 and 4, the horizontal axis identifies different Δt values, and the vertical axis identifies the performance metrics. The blue column ($N=0$ and $k=0$) means the traditional popularity is used and the attribute- and trend-aware components are missing. The popularity for the orange column ($N=4$ and $k=0$) and gray column ($N=0$ and $k=3$) is equipped with the attribute-aware and trend-aware components, respectively. The popularity used for the red column ($N=4$ and $k=3$) holds both the attribute-aware and trend-aware components. The green horizontal line denotes the benchmark performance obtained with the

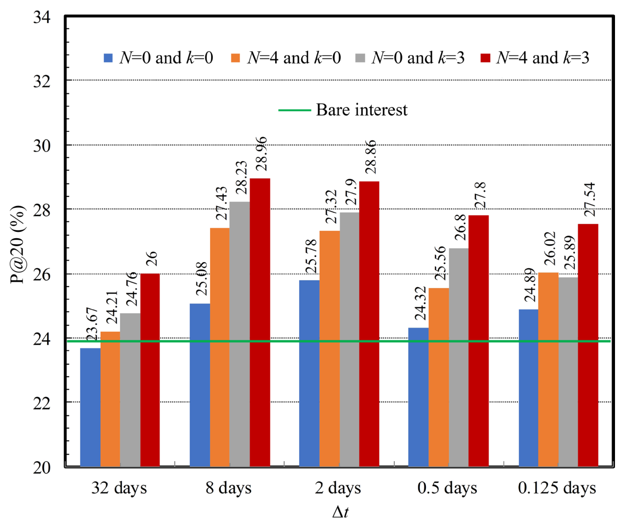


Fig. 3 On Movieslen-1m, the P@20 with varying PID configurations (Δ*t*, *N*, and *k*).

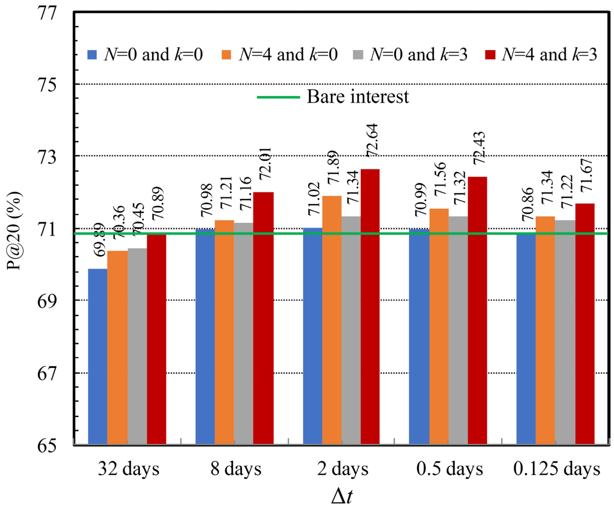


Fig. 4 On Yoochoose1/64, the P@20 with varying PID configurations (Δt **,** N **, and** k **).**

bare interest (regardless of popularity).

For vertical comparison, using interest enhanced by the traditional popularity ($N=0$ and $k=0$) for recommendation only slightly improves performance, especially in the Yoochoose1/64 dataset. This suggests that traditional popularity is weak in the face of temporal datasets or scenarios with high popularity requirements. However, adding either the attribute-aware component or the trend-aware component has resulted in dramatically advanced recommendation results, which suggests that attributes and trends help to achieve more accurate popularity in dynamic datasets. In addition, in Movieslen-1m, the performance gain from trend-aware components is generally higher than that from attribute-aware components, while in Yoochoose1/64, it is just the opposite. That may be because, in the movie scenario, the popularity trend of the movie has a great impact on its current popularity. Most users usually choose movies based on past trends. However, in the e-commerce scene, there are so many mass consumer goods that distinguishing the attributes of items is particularly important for accurate popularity. Moreover, in Movieslen-1m, the optimal performance is obtained under the conditions of Δt = 8 days, $N=4$, and $k=3$, that is, the relevant records from the past 32 days are considered to refine popularity. However, in Yoochoose1/64, the optimal conditions are in Δt = 8 days, $N=4$, and $k=3$, which means that only records from the last 8 days need to be considered. This is because a movie usually becomes popular after a long period of public commentary, while the popularity in the e-commerce scenario responds more quickly to potential factors. The time span for most items to be popular is about a week.

For horizontal comparison, in any time period in the past, only considering the PR value is much weaker than carefully making sense of its trend. For example, in the past 32 days, on Movieslen-1m, P@20 of the model with the traditional popularity (Δt = 32 days, $N=0$, and $k=0$) is 23.67%, and P@20 is 28.96% when setting Δt to 8 days, N to 4, and k to 3. Note that this conclusion is consistently tenable at various time in the past. Therefore, in a fixed period of time, the popularity extracted by PID is also superior to traditional popularity only considering an absolute proportion.

Vertical and horizontal comparisons demonstrate that the attribute-aware component and the trend-aware component are effective and necessary for accurate popularity. Therefore, the proposed attribute-aware and trend-aware popularity are superior to traditional ratios and metrics. Likewise, the second question has been answered.

5.3 Mechanism for improving recommendation results by popularity-aware interest

As aforementioned, the popularity-aware interest improves the recommendation results in a beyond-graph topology manner. To understand the mechanism more clearly, the experiments are conducted based on SRGNN and TAGNN. In the original SRGNN and TAGNN, the linear session sequences are processed into session graphs. Two general adjacency matrices A_G^{O} and A_G^{I} depict the outgoing and incoming adjacency relationships (pairwise transitions) and can be defined as

$$A_G^{\mathrm{O}}[i,j]=\delta_{i,j}^{G-\mathrm{out}} \tag{16}$$

$$A_G^{\mathrm{I}}[i,j]=\delta_{i,j}^{G-\mathrm{in}} \tag{17}$$

$$\delta_{i,j}^{G-\mathrm{out}}=\begin{cases}\dfrac{\mathrm{sum}(v_i,v_j)}{\sum_{v_k\in N_{\mathrm{out}}(i)}\mathrm{sum}(v_i,v_k)}, & i\neq j;\\ 1, & i=j\end{cases} \tag{18}$$

$$\delta_{i,j}^{G-\mathrm{in}}=\begin{cases}\dfrac{\mathrm{sum}(v_j,v_i)}{\sum_{v_k\in N_{\mathrm{in}}(i)}\mathrm{sum}(v_k,v_i)}, & i\neq j;\\ 1, & i=j\end{cases} \tag{19}$$

where $\mathrm{sum}(v_i,v_j)$ is defined as the number of occurrences that item v_j is interacted with by the user after item v_i, namely, the count of (v_i,v_j) edge. $N_{\mathrm{out}}(i)$ and $N_{\mathrm{in}}(i)$ represent the set of successor and predecessor nodes of v_i, respectively. Because the self-effect of nodes is of significance to the graph structure, the $\delta_{i,i}^{G-\text{out}}$ and $\delta_{j,j}^{G-\text{in}}$ are set to 1.

To improve recommendation performance from the session graph perspective, we refine the pairwise transition between items by adding the time interval. This is because accurately capturing the transitions between items helps in better item representation and further leads to a more accurate representation of session interest. Generally, the shorter the time interval, the higher the correlation between two sequential items. Therefore, the weight of the edges is inversely proportional to the time interval[30]. If the item v_j is interacted with after item v_i, the time interval is constructed as

$$\delta_{i,j}^{t} = \begin{cases} \mathrm{e}^{-\frac{t_{v_i,v_j}}{t_{\max}}}, & i \neq j; \\ 1, & i = j \end{cases} \tag{20}$$

where $t_{\max}$ is the max of the time interval of all edges in session graphs, and thereby $t_{v_i,v_j}\big/t_{\max}$ denotes the normalized time interval. Furthermore, two time-aware adjacency matrices $\boldsymbol{A}_T^{\mathrm{O}}$ and $\boldsymbol{A}_T^{\mathrm{I}}$ are built as follows:

$$\boldsymbol{A}_G^{\mathrm{O}}[i,j] = \delta_{i,j}^{t} \tag{21}$$

$$\boldsymbol{A}_G^{\mathrm{I}}[i,j] = \delta_{j,i}^{t} \tag{22}$$

The final connection matrix is shown as

$$\boldsymbol{A}_s = [\boldsymbol{A}_G^{\mathrm{O}} \| \boldsymbol{A}_G^{\mathrm{I}} \| \boldsymbol{A}_T^{\mathrm{O}} \| \boldsymbol{A}_T^{\mathrm{I}}] \tag{23}$$

In this way, $\boldsymbol{A}_s$ quantifies the transitions between an item and its neighbors according to the time interval. By constructing $\boldsymbol{A}_s$, we expect to improve recommendation performance by more accurately depicting the transition relationship between items.

Finally, four variants are built in detail based on SRGNN and TAGNN. The variants about time interval highlight precisely capturing the transitions of items, while variants about popularity aim to construct popularity-aware interest for matching in a beyond-graph topology manner.

- TAGNN (T.exl.) is a variant of TAGNN. It uses $\boldsymbol{A}_s$ to capture the more accurate transition relationship of intra-session items.
- TAGNN (po.exl.) is a variant of TAGNN. It uses the popularity-aware interest to recommend and aims to improve recommendations in a beyond-graph topology manner.
- SRGNN (T.exl.) is a variant of SRGNN. It uses $\boldsymbol{A}_s$ to capture the more accurate transition relationship of intra-session items.
- SRGNN (po.exl.) is a variant of SRGNN. It uses the popularity-aware interest to recommend and aims to improve recommendations in a beyond-graph topology manner.

The results on TAGNN and SRGNN are shown in Tables 3 and 4, where the relative performance improvements are shown in parentheses and the highest performance is highlighted in boldface. Surprisingly, it can be observed that both time interval information and the popularity-aware interest can improve P@20 and MRR@20, but statistically, the advantage of the time interval is mainly reflected in the MRR@20, and yet the popularity-aware interest shows advantages in terms of P@20. We argue that the time interval is more conducive to refining and analyzing the correlation between different items of sessions, so as to gain

Table 3 Performance of contrast variants of TAGNN.

Variant	Yoochoose1/64		Movieslen-1m	
	P@20 (%)	MRR@20 (%)	P@20 (%)	MRR@20 (%)
TAGNN	70.85 (–)	30.78 (–)	24.11 (–)	7.60 (–)
TAGNN (T.exl)	71.18 (0.47↑)	**32.35 (5.10↑)**	25.51 (5.81↑)	**9.22 (21.31↑)**
TAGNN (po.exl)	**72.64 (2.53↑)**	(31.85) (3.48↑)	**28.96 (20.12↑)**	9.06 (19.21↑)

Table 4 Performance of contrast variants of SRGNN.

Variant	Yoochoose1/64		Movieslen-1m	
	P@20 (%)	MRR@20 (%)	P@20 (%)	MRR@20 (%)
SRGNN	70.57 (–)	30.94 (–)	23.00 (–)	7.01 (–)
SRGNN (T.exl)	70.74 (0.24↑)	**31.88 (3.04↑)**	23.37 (1.61↑)	**8.34 (18.97↑)**
SRGNN (po.exl)	**71.87 (1.84↑)**	(31.32) (1.23↑)	**26.57 (15.52↑)**	7.88 (11.04↑)

insight into the user's behavior and preferences. The accurate preference helps to put the ground truth at the top of the recommendation list, thereby improving MRR. However, popularity-aware interest considers popularity as a source of interest. It serves as another standard to crawl more similar items to the recommendation lists and further increases the probability that the ground truth is included in the recommendation list, leading to a higher P@20. In addition, this phenomenon also reflects that users' interest is easily affected by popularity, and it is reasonable to construct popularity-aware interest.

6 Conclusion

6.1 Theoretical and practical implications

In this paper, we aim to drive popularity extraction and propose a general strategy to consistently improve the performance of SBR. Unlike the traditional methods representing the popularity only by a simple ratio or metric (e.g., a ratio of users who have rated the item and the box office of a movie), we further add the attribute- and trend-aware components. The introduction of attribute- and trend-aware components is guided by the PID idea in the automatic control field and thus is an interdisciplinary application. In addition, the bare interest (without considering the other factors that are sensitive to the session interest) is one of the obstacles to further improve the performance of SBR. To alleviate this problem, we use the popularity of the last items to enhance the session interest modeling, called popularity-aware interest. Meanwhile, this is also a rare combination of popularity and the SBR.

Comprehensive experiments performed on two real-world datasets show the effectiveness of our proposed popularity-aware interest and the superiority of attribute- and trend-aware popularity to traditional popularity. Further comparison shows the mechanism by which the popularity-aware interest improves recommendation results. The popularity is considered as a source of interest to enrich the semantics of the session interest. It serves as another standard to crawl more similar items to the recommendation lists and further increases the probability that the ground truth is included in the recommendation list, leading to improved performance. All the above insights broaden our understanding for the popularity concept and the task of SBR, which contributes to the entire recommendation community.

In practice, our effort is also of practical significance. First, popularity is an important feature in many scenarios, especially in movie recommendation and e-commerce platforms. In extended studies, popularity is fundamental. Therefore, advanced popularity helps effectively solve a number of practical problems (e.g., cold start and popularity bias) and brings indirect benefits. In addition, the popularity-aware interest takes the consumer psychology of most people into account and is a coexistence of popularity-based recommendation philosophy and personalized recommendations philosophy, thus improving the accuracy of user behavior prediction and the consumption experience and increasing revenue.

6.2 Summary and future work

In this paper, trend- and attribute-aware popularity is extracted according to the PID idea. It is applied in SBR through an entry point of the user's interest. The resulting performance demonstrates the superiority of trend- and attribute-aware popularity over traditional popularity. Moreover, popularity-aware interest as a strategy consistently improves the performance of the main methods. For STAMP, SRGNN, GCSAN, and TAGNN, on Yoochoose1/64, the metric P@20 is relatively improved by 0.93%, 1.84%, 2.02%, and 2.53%, respectively and MRR@20 is relatively improved by 3.74%, 1.23%, 2.72%, and 3.48%, respectively. On Movieslen-1m, the relative improvements of P@20 are 7.41%, 15.52%, 8.20%, and 20.12%, and that of MRR@20 are 2.34%, 12.41%, 20.34%, and 19.21%, respectively.

In the future, we plan to continue to enrich the semantics of item popularity. There may be other factors dominating the popularity of items in a specific scenario (e.g., the popularity of a song is influenced by the popularity of a singer). In addition, we will mine new scenarios for popularity and apply popularity to dynamic scenarios, contributing consistently to the recommendation community.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 62276278) and Guangdong Basic and Applied Basic Research Foundation (No. 2022A1515110006).

References

[1] Y. Ma, J. Mao, Z. Ba, and G. Li, Location

recommendation by combining geographical, categorical, and social preferences with location popularity, *Inf. Process. Manag.*, vol. 57, no. 4, p. 102251, 2020.

[2] E. Yalcin, Blockbuster: A new perspective on popularity-bias in recommender systems, in *Proc. 2021 6th Int. Conf. Computer Science and Engineering* (*UBMK*), Ankara, Turkey, 2021, pp. 107–112.

[3] Z. Zhao, J. Chen, S. Zhou, X. He, X. Cao, F. Zhang, and W. Wu, Popularity bias is not always evil: Disentangling benign and harmful bias for recommendation, *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 10, pp. 9920–9931, 2023.

[4] J. Oh, S. Park, H. Yu, M. Song, and S. T. Park, Novel recommendation based on personal popularity tendency, in *Proc. 2011 IEEE 11th Int. Conf. Data Mining*, Vancouver, Canada, 2011, pp. 507–516.

[5] H. Abdollahpouri, M. Mansoury, R. Burke, and B. Mobasher, The connection between popularity bias, calibration, and fairness in recommendation, in *Proc. 14th ACM Conf. Recommender Systems*, Virtual Event, 2020, pp. 726–731.

[6] H. Steck, Item populartity and recommendation accucy, in *Proc. 5th ACM Conf. Recommender Systems*, Chicago, IL, USA, 2011, pp. 125–132.

[7] S. Ekinci, B. Hekimoğlu, and D. Izci, Opposition based Henry gas solubility optimization as a novel algorithm for PID control of DC motor, *Eng. Sci. Technol. Int. J.*, vol. 24, no. 2, pp. 331–342, 2021.

[8] L. Wang, *PID Control System Design and Automatic Tuning Using MATLAB/Simulink*. Chichester, UK: John Wiley & Sons Ltd, 2020.

[9] A. Li, Z. Cheng, F. Liu, Z. Gao, W. Guan, and Y. Peng, Disentangled graph neural networks for session-based recommendation, *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 8, pp. 7870–7882, 2023.

[10] H. Abdollahpouri and R. Burke, Multistakeholder recommender systems, in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, eds. New York, NY, USA: Springer, 2022, pp. 647–677.

[11] F. Wei, S. Chen, J. Jin, S. Zhang, H. Zhou, and Y. Wu, Adaptive alleviation for popularity bias in recommender systems with knowledge graph, *Secur. Commun. Netw.*, vol. 2022, p. 4264489, 2022.

[12] M. Bressan, S. Leucci, A. Panconesi, P. Raghavan, and E. Terolli, The limits of popularity-based recommendations, and the role of social ties, in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 745–754.

[13] I. Barjasteh, R. Forsati, D. Ross, A. H. Esfahanian, and H. Radha, Cold-start recommendation with provable guarantees: A decoupled approach, *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 6, pp. 1462–1474, 2016.

[14] J. Wang, Y. Chen, Z. Wang, and W. Zhao, Popularity-enhanced news recommendation with multi-view interest representation, in *Proc. 30th ACM Int. Conf. Information & Knowledge Management*, Virtual Event, 2021, pp. 1949–1958.

[15] A. K. Singh and K. Lakshmanan, PILHNB: Popularity, interests, location used hidden Naive Bayesian-based model for link prediction in dynamic social networks, *Neurocomputing*, vol. 461, pp. 562–576, 2021.

[16] H. J. Ahn, Utilizing popularity characteristics for product recommendation, *Int. J. Electron. Commer.*, vol. 11, no. 2, pp. 59–80, 2006.

[17] R. Kumar, B. K. Verma, and S. S. Rastogi, Social popularity based SVD++ recommender system, *Int. J. Comput. Appl.*, vol. 87, no. 14, pp. 33–37, 2014.

[18] Y. Zhang, F. Feng, X. He, T. Wei, C. Song, G. Ling, and Y. Zhang, Causal intervention for leveraging popularity bias in recommendation, in *Proc. 44th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Virtual Event, 2021, pp. 11–20.

[19] M. Kumar, S. Srivastava, and N. Uddin, Forgery detection using multiple light sources for synthetic images, *Aust. J. Forensic Sci.*, vol. 51, no. 3, pp. 243–250, 2019.

[20] C. Dhasarathan, M. Kumar, A. K. Srivastava, F. Al-Turjman, A. Shankar, and M. Kumar, A bio-inspired privacy-preserving framework for healthcare systems, *J. Supercomput.*, vol. 77, no. 10, pp. 11099–11134, 2021.

[21] D. Tiwari, B. Nagpal, B. S. Bhati, A. Mishra, and M. Kumar, A systematic review of social network sentiment analysis with comparative study of ensemble-based techniques, *Artif. Intell. Rev.*, doi: 10.1007/s10462-023-10472-w.

[22] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, Parallel recurrent neural network architectures for feature-rich session-based recommendations, in *Proc. 10th ACM Conf. Recommender Systems*, Boston, MA, USA, 2016, pp. 241–248.

[23] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, Neural attentive session-based recommendation, in *Proc. 2017 ACM on Conf. Information and Knowledge Management*, Singapore, 2017, pp. 1419–1428.

[24] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang, STAMP: Short-term attention/memory priority model for session-based recommendation, in *Proc. 24th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, London, UK, 2018, pp. 1831–1839.

[25] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, Session-based recommendation with graph neural networks, *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 346–353, 2019.

[26] C. Xu, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, F. Zhuang, J. Fang, and X. Zhou, Graph contextualized self-attention network for session-based recommendation, in *Proc. 28th Int. Joint Conf. Artificial Intelligence*, Macao, China, 2019, pp. 3940–3946.

[27] F. Yu, Y. Zhu, Q. Liu, S. Wu, L. Wang, and T. Tan, TAGNN: Target attentive graph neural networks for session-based recommendation, in *Proc. 43rd Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Virtual Event, 2020, pp. 1921–1924.

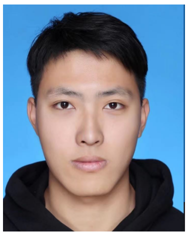
[28] P. K. Guchhait and A. Banerjee, Stability enhancement of wind energy integrated hybrid system with the help of static synchronous compensator and symbiosis organisms search algorithm, *Prot. Contr. Mod. Power Syst.*, vol. 5, no. 1, pp. 1–13, 2020.

[29] Z. Gao, A tuning method via Borges derivative of a neural network-based discrete-time fractional-order PID controller with Hausdorff difference and Hausdorff sum, *Fractal Fract.*, vol. 5, no. 1, p. 23, 2021.

[30] W. Chen, P. Ren, F. Cai, F. Sun, and M. D. Rijke, Improving end-to-end sequential recommendations with intent-aware diversification, in *Proc. 29th ACM Int. Conf. Information & Knowledge Management*, Virtual Event, 2020, pp. 175–184.

Yuze Peng received the BEng degree in optoelectronics science and technology from Shenzhen University, Shenzhen, China in 2016 and the MEng degree in electronic science and technology from Nankai University, Tianjin, China in 2019. He is currently pursuing the PhD degree at School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China. He is committed to using Field Programmable Gate Array (FPGA) to accelerate specific neural network models and applying.

Qingkun Chen received the BEng degree in microelectronics and the MEng degree in microelectronics and solid-state electronics from Fuzhou University, Fujian, China in 2015 and 2018, respectively, and the PhD degree from Sun Yat-sen University, Guangzhou, China in 2022. His research interests include FPGA acceleration and network-on-chip design.



Shengjun Xu received the BEng degree in electrical engineering and its automation from Jiangsu University, Zhenjiang, China in 2020. He is currently pursuing the MEng degree at School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China. He is committed to using FPGA to accelerate specific neural network models and applying.

Yihua Huang received the BEng degree in mechanical engineering from Nanjing University of Science and Technology, Nanjing, China in 1996, the MEng degree in aerospace propulsion engineering from Nanjing University of Science and Technology, Nanjing, China in 1999, and the PhD degree from University of Science and Technology of China, Hefei, China in 2002. From 2002 to 2013, he was an assistant and associate professor at School of Electronic and Communication Engineering, Sun Yat-sen University, Guangzhou, China. From 2013, he has been a professor at School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China. His research interests include deep neural network acceleration on FPGA and system design on FPGA.

Wenjin Huang received the BEng degree in automation and the MEng degree in circuits and system from Sun Yat-sen University, Guangzhou, China in 2012 and 2015, respectively, and the PhD degree from Sun Yat-sen University, Guangzhou, China in 2021. He is currently doing postdoctoral research at Sun Yat-sen University, Guangzhou, China. His research interests include high performance computing and FPGA acceleration for DNN models.