Front. Comput. Sci., 2023, 17(5): 175612
https://doi.org/10.1007/s11704-022-2223-1

RESEARCH ARTICLE

Quantifying predictability of sequential recommendation via logical constraints

En XU[1], **Zhiwen YU (✉)**[1], **Nuo LI**[1], **Helei CUI**[1], **Lina YAO**[2], **Bin GUO**[1]

1 School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China
2 School of Computer Science and Engineering, University of New South Wales, Sydney NSW 2052, Australia

Abstract The sequential recommendation is a compelling technology for predicting users' next interaction via their historical behaviors. Prior studies have proposed various methods to optimize the recommendation accuracy on different datasets but have not yet explored the intrinsic predictability of sequential recommendation. To this end, we consider applying the popular predictability theory of human movement behavior to this recommendation context. Still, it would incur serious bias in the next moment measurement of the candidate set size, resulting in inaccurate predictability. Therefore, determining the size of the candidate set is the key to quantifying the predictability of sequential recommendations. Here, different from the traditional approach that utilizes topological constraints, we first propose a method to learn inter-item associations from historical behaviors to restrict the size via logical constraints. Then, we extend it by 10 excellent recommendation algorithms to learn deeper associations between user behavior. Our two methods show significant improvement over existing methods in scenarios that deal with few repeated behaviors and large sets of behaviors. Finally, a prediction rate between 64% and 80% has been obtained by testing on five classical datasets in three domains of the recommender system. This provides a guideline to optimize the recommendation algorithm for a given dataset.

Keywords sequential recommendation, information theory, predictability

1 Introduction

Recommender systems have been in full swing recently and have played a significant role in all aspects of people's lives. Accurate user interest modeling enhances user experience and also boosts company profits. As shown in Fig. 1, various methods of sequential recommendation [1–3] are emerging, and the accuracy is improving. This inspires us to think about a question, that is, what highest accuracy can be achieved in the assumption of designing an optimal model for the sequential recommendation of a given dataset?

The highest accuracy that can be achieved on a dataset is defined as predictability (Π). The data collected by sequential recommendation is sequences formed from the interaction records between users and items. The most widely influential research on the predictability of sequential data comes from [4] on human mobility. To quantify the entropy of movement sequence and calculate predictability, Song et al. propose the corresponding theory. By quantifying the entropy in the mobile behavior of 50,000 individuals in three months, it is found that the predictability of user mobility is 93%. The theory does not assume anything and has good generalizability. It has been widely used to measure the predictability of various types of sequences. We apply this theory to sequential recommendations to estimate its predictability in this work.

The predictability of sequential recommendation can allow us to understand the extent to which a user's next interaction can be predicted and the regularity of user behavior. Predictability will enable us to quickly know the highest accuracy achievable for a dataset and determine how difficult it is for the dataset to be predicted accurately. This saves more time and effort than reproducing the algorithm and deriving the accuracy. Predictability is an inherent property of data that enhances our knowledge of it. Understanding the gap between the accuracy achieved by existing algorithms and the predictability can show us how much room there is for improvement in today's sequential recommendation algorithms.

To explore the predictability of sequential recommendation, we draw on the predictability theory of human mobile behavior. However, a direct application of the approach to sequential recommendation can lead to serious deviations. The behavior candidates' size (N) is an essential metric in the predictability calculation. Applying existing methods for computing N to sequential recommendations leads to serious biases and yields incorrect predictability. One of the existing methods for calculating N is to directly count the number of different items the user has purchased [4], and the other is to estimate the maximum number of different items purchased immediately after the user has purchased an item [5]. If one thinks in terms of graphs, the previous one counts the number

Received April 14, 2022; accepted July 20, 2022

E-mail: zhiwenyu@nwpu.edu.cn

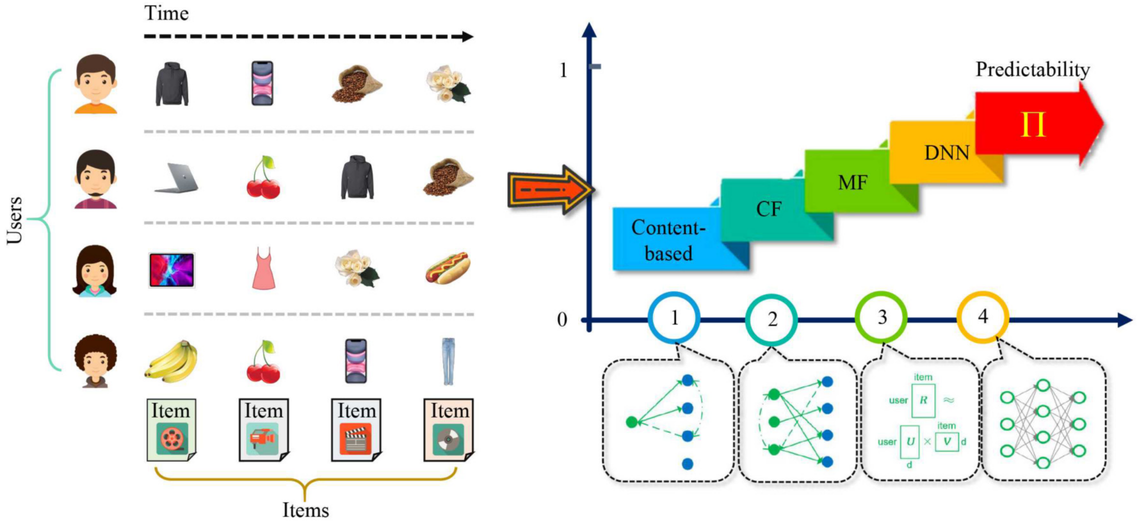

Fig. 1 Schematic representation of the predictability of sequential recommendation. As new algorithms continue to be proposed, the accuracy of sequential recommendations continues to improve. Predictability is the potential maximum accuracy

of nodes, and the latter calculates the maximum degree of the nodes. Both methods show significant deviations in recommendations when there is little user behavior. They're all underestimates. When there is little user behavior in the recommendation, it does not mean that the user's behavior is very particular. On the contrary, when there is insufficient data, the user has more choices, and it's more challenging to determine the next interaction. The existing methods in other scenarios also show significant deviations, which we describe in detail in Section 3.5.

The N of the movement sequences are constrained based on topological relationships in geography. Our study found that although the behavior in sequential recommendation is not spatially constrained, it still has logical constraints. We use global historical data to learn the logical connections between items. The user's following location in mobile behavior selects sites that are spatially close to historical locations, while the user in recommendation chooses items that are logically similar to historical items. We propose method M_{lc} to screen N using logical constraints between candidate items and historical data. This method is straightforward to implement. Based on method M_{lc}, we propose the method M_{bpaa} (best prediction algorithm available). We derive the N by constructing the relationship between the *Top-N* accuracy and N. The main contributions of our paper are as follows:

- We propose an approach to estimate the predictability of sequential recommendation better to understand the development levels of current recommender systems and provide insights into how much room to improve the accuracy.
- We excavate the logical relationship between items and find that user behaviors will be aggregated in the logical space. We then use the logical constraint between the user's next behavior and the historical behaviors to achieve candidate set screening.
- We define the problem of quantifying N in the sequential recommendation and turn it into two tasks. One is how to use historical data to predict N, and the other is how large N can determine user behavior given historical data. We propose two methods for computing N, one easier to implement and the other more accurate.

2 Related work

2.1 Sequential recommendation

The traditional models to solve the sequential recommendation problem include sequential pattern mining [6,7] and Markov chain model [8], which are the most intuitive ways to view user behavior from sequential characteristics. Sequential pattern mining uses data mining to find frequent items that meet a specific pattern to get users' shopping habits. It is conceivable that the method of mining user behavior based on this simple model is not good enough. This method can only dig out some frequent patterns. How to use the relationship between infrequent behaviors and how to mine the complex relationship between behaviors are the problems that cannot be solved by this method.

Due to its strong nonlinear fitting ability, The RNN is a classic method of capturing sequential information in deep learning [2], so the first batch of sequential recommendation models based on deep learning also used this method. Later, a series of RNN-based development methods, such as LSTM [9] and Bi-LSTM [10] were used. Other deep learning methods, such as CNN and GNN, have also been applied to sequential recommendations due to their advantages. Given a series of user-item interactions, the CNN-based method first puts the embeddings of all these interactions into a matrix and then treats such a matrix as an "image" [11]. CNN uses convolution filters for recommendation and learns sequence patterns as local features of the image. Due to the powerful ability of GNN to process graphs [12], it can convert user behavior sequences into directed graphs to learn complex relationships in structured datasets.

2.2 Predictability

The theory of predictability is to analyze the entropy in the data from the perspective of information theory, then construct the association between predictability and entropy, and finally derive the predictability of the system through entropy. This theory was first proposed by Song et al. in Science [4]. It used information entropy to quantify the chaos of human movement data and calculated that the predictability of human movement behavior reached 93% by scaling Fano's inequality. Since then, many studies have been conducted using their formulas on different domain datasets, some directly, while others have been further extended to improve the theory. Some researchers have tried various methods or improved entropy measures to quantify the predictability of human activities, such as mutual information [13] and instantaneous entropy [14,15]. After that, Smith et al. integrated the topological constraints in the real world into calculating the upper bound of the predictability of movement behavior [5]. This improvement considers that certain positions are simply inaccessible for the next move, thus providing a significantly tighter upper limit and thus yielding more accurate predictability.

2.3 Predictability of sequential recommendation

There is only a little literature on the predictability of sequential recommendation. Krumme and Nguyen directly applied the theory of the predictability of mobile behavior in the recommendation and combine it with different data generated by users in shopping scenes to obtain the predictability of shopping behavior [16,17]. These methods do not consider the significant bias introduced by the direct use of [4]. There are attempts to analyze the upper bound on its accuracy for a specific recommendation algorithm. Zhang et al. obtain the upper limit of its recommended accuracy by calculating the limitation that the diffusion algorithm can propagate [18]. Similarly, for several classic recommendation algorithms, Järv can directly determine the recommended items at the next moment through statistical analysis and can roughly estimate the upper limit of the algorithm's accuracy after observing the distribution of the data [19]. But these simple rule-based methods only yield very rough predictability.

3 Predictability of sequential recommendation

This section will formally define the predictability of sequential recommendation and introduce in detail how to calculate the predictability for a given dataset.

3.1 The definition of predictability

Predictability (Π): Given the user behavior sequence $T = \{b_1, b_2, ..., b_n\}$, considering the randomness and the relevance of the user behaviors, the highest accuracy that can be achieved with this dataset. Upper bound of predictability (Π^{up}): The upper bound on the predictability corresponding to the dataset. Our goal is to obtain as small an upper bound as possible.

Sequential recommendation: In sequential recommendation, let $U = \{u_1, u_2, ..., u_{|u|}\}$ represent a group of users, $V = \{v_1, v_2, ..., v_{|v|}\}$ is a group of items, and list $B_u = [v^u_1, ..., v^u_t, ..., v^u_{n_u}]$ to represent the user's $u \in U$ interaction sequence in chronological order, where $v^u_t \in V$ is the item that user u interacts with at time step t, and n_u is the length of the user's interaction sequence. Given the interaction history B_u, the sequential recommendation aims to predict the items that the user u will interact with at time step $n_u + 1$. It can be formalized as modeling probability:

$$p(v^u_{n_u+1} = v \mid B_u). \tag{1}$$

The upper bound of predictability Π^{up} of sequential recommendation is the highest accuracy that the best method can achieve on the data given the user interaction history B_u.

3.2 Entropy

Entropy is an effective way to measure the chaos in a system. The higher the entropy is, the more chaotic the system is. The lower the entropy is, the more ordered the system is. Generally, low entropy means higher predictability.

$$S_{real} = -\sum_{T' \in T} P(T')\log_2[P(T')]. \tag{2}$$

The real entropy S_{real} [20] depends not only on the frequency of items but also on the order in which the item is purchased to capture all the information existing in a user's behaviors. $T = \{b_1, b_2, ..., b_n\}$ represents the user behavior sequence, and the real entropy is calculated as the Eq. (2). Among them, $P(T')$ represents the probability of finding a specific time-ordered subsequence T' in the sequence T. From the formula, we can know that in the T, if the same behavior b_i occurs at different times, the entropy contained in them is different. Therefore, it is well suited to measure the entropy in sequence.

The problem of finding all subsets of a given set has exponential complexity ($O(2^n)$). Here, we use the Lempel-Ziv [21] estimator to calculate the real entropy. The Lempel-Ziv estimator can quickly converge to the real entropy. For the sequence of user behavior after time n, the entropy can be estimated in the following way:

$$S^{est} = \left(\frac{1}{n}\sum_i \Lambda_i\right)^{-1} \ln n. \tag{3}$$

Here, Λ_i represents the length of the shortest subsequence that has never appeared from 1 to $i-1$ starting from time i.

3.3 Upper bound for predictability Π^{up}

Π represents the predictability of sequential recommendation, while Π^{up} represents an upper bound on the predictability of sequential recommendation. Song et al. [4] use Fano's inequality scaling to prove that $\Pi \leqslant \Pi^{up}$, and Π^{up} satisfies the following formula:

$$\begin{aligned} S &= -\Pi^{up}\log_2\Pi^{up} - (1-\Pi^{up})\log_2(1-\Pi^{up}) \\ &\quad + (1-\Pi^{up})\log_2(N_s - 1). \end{aligned} \tag{4}$$

Among them, the upper bound for predictability Π^{up} is a value between 0 and 1. In theory, the higher the value is, the more regular the user's behavior is. The S is the actual entropy calculated based on the sequence of user behavior. The N_s represents the number of items that this user is likely to interact with at the next moment. The S and N_s corresponding to the sequence are calculated, and the predictability is obtained by bringing them into the Eq. (4). Table 1 organizes several important symbols in this paper.

Table 1 Meanings of main symbols used

Symbol	Meaning
N_s	N calculated by method M_s
N_r	N calculated by method M_r
N_{lc}	N calculated by method M_{lc}
N_{bpaa}	N calculated by method M_{bpaa}
Π	Predictability
Π^{up}	The upper bound of predictability

3.4 Existing method of calculating N

Inspired by the predictability of human mobile behavior, many scenarios use this theory to study predictability to obtain results on the predictability of different types of data. The predictability analysis framework of human movement behavior can be extended to other types of time series, such as the sequence of interpersonal communication [13,22,23], vehicle mobility [24–28], and the sequence of IP addresses for cyber attacks [29], stock price changes [30], electronic health records [31]. Some studies directly apply the theory to the predictability calculation of sequential recommendations. However, direct application is not appropriate, which was not considered in the previous papers. From the Eq. (4), we can see that to calculate the predictability of the sequence. The focus is to obtain two values, namely S and N. From the knowledge of information theory, we can see that the S can be directly calculated by the Eq. (3). For the calculation of N, Song et al. [4] were to observe how many different places the user has visited in the historical data. The number of places the user is likely to go to is obtained from this. We call this method M_S. The calculation of the N is mathematically formalized as follows:

$$T = \{b_1, b_2, ..., b_n\}, \tag{5}$$

$$N_s = |\{x \mid x \in T\}|, \tag{6}$$

where T represents a sequence of human behavior trajectories.

Smith et al. [5] gave a more accurate upper limit of the number of locations that users may visit at the next moment through topological constraints. We call this method M_r. The author claimed that although humans may visit all the locations in the T collection at the next moment, in fact, due to geographical constraints, humans cannot visit all the locations in the next moment, so the calculation by the Eq. (7) can get more accurate upper limit:

$$N_r = \max_{x\in\Omega} |\{s_{i+1} : s_i = x\}|. \tag{7}$$

This is an overestimation of the reachability of each data drive. Importantly, this data-driven approach prevents overestimating possible "next step" positions. The core of the advanced method is to let us focus not on the number of locations that users may visit under global time but on the next moment. This is more in line with the theoretical model of predictability.

3.5 Shortcomings of existing methods

In sequential recommendation, since users will often buy items they have never purchased before and rarely buy duplicate items, these two methods of computing N can be highly biased. If the user behavior sequence is $h = \{b_1, b_2, ..., b_m\}$, the N calculated by M_s is M, and the N calculated by M_r is 1. The meaning of N is the size of the user's candidate set at the next moment. When the sequence is short, we assume that $M = 5$. It cannot be shown that the user will only select from these 5 items in the next moment. Conversely, the set of items that users are likely to interact with is large because not enough user data is collected, and the entropy of the next moment of behavior is high. When M is huge, the result of method M_s will be huge, and the result of method M_r is tiny, both deviating from the actual situation. The two existing methods work because users do not constantly interact with new items. Assuming $N = 100$ and that users do not interact with new items, we can finally get the set of users' next interactions if we count the data long enough. Suppose $N = 100$, but the user is constantly interacting with new items. In that case, the calculated N will keep increasing as the length of the collected data increases, but the real N does not change. Therefore, obtaining N based on simple rules is unsuitable for the sequential recommendation. We need to use the association of behaviors in the temporal order to constrain the next-moment behavior. *This problem translates into using historical data to predict the size of the set of user's next interactions.*

Observing the Eq. (4), we will find that the inaccurate estimation of N will lead us to unreal predictability. In extreme cases, when N tends to infinity, the predictability tends to 1. This suggests that user behavior can be accurately predicted, which is a serious deviation from reality. A reasonable N estimation is especially important for us to calculate accurate predictability.

4 Our approaches

4.1 Details of method M_{lc}

Method motivation: We need to predict N using historical data, so we mine the association of historical behaviors with the immediately occurring behaviors to qualify N. Mobile behavior data is a constant repetition of the user's historical behavior. Therefore the metrics directly from the historical data can reflect N well. The sequential recommendation requires digging deeper associations between items to find the relationships between the actions before and after. We use the graph learning method to get the connection between items and then statistically find that the similarity between user history items is significantly higher than the similarity between general items. Based on this conclusion, we find a way to limit the candidate items to the next moment.

Due to the topological constraints of human movement behavior, the trajectory will be clustered around an area, covering a limited number of locations. Although there are no topological constraints in the sequential recommendation scenario, our research has found logical constraints between user behaviors. First, we learn to get the global items' association relationship and get the similarity between the items. Then it is further discovered that the similarity between the user's historical behavior is significantly higher than the general similarity between the global items. This shows that the user's behavior is not random and is highly regular from the level of logical association. The similarity between the user's historical behaviors decreases as the distance between the behaviors increases. This is also in line with our intuition. We are based on a tight correlation between historical data and the next interaction, thus enabling the filtering of N. We propose the method M_{lc}（Algorithm 1） to calculate N, as shown below:

```
Algorithm 1 Method logical constraints
Require: $HP_l$, $HP_r$, $HP_s$
Ensure: Candidate set size $N$
1: Select 10 ($HP_l$) items that the user has recently interacted with
2: Find those items that are more similar to 60% ($HP_r$) in 10 ($HP_l$) items than 0.7 ($HP_s$) in the global items
3: Calculate the number of items that meet the above conditions, and get the $N$
```

The above parameters are obtained from our experiments. The specific values of each parameter under different datasets are shown in Table 2. These parameters are a reflection of the closeness of the user's historical items, while the similarity between two random items does not satisfy the above requirements. We will make a detailed description in the experimental part, Sections 5.2 and 5.3.

Item association relationship mining: After having user behavior sequence data, we first need to learn to obtain the association between items. So that we can further explore the rules of user behavior, we can directly use the number of common occurrences between items to indicate the similarity between items. The more common events, the higher the similarity between the two items. However, this method does not work well and cannot accurately reflect the correlation between items. At the same time, if this method is adopted, we need to store a large table for the item to obtain the similarity between an item and other items. Here, we borrow word2vec [32] from the work of natural language processing. We regard the user's historical behavior sequence as a language sample and learn embedding by using the Skip-Gram algorithm, which will maximize the simultaneous appearance probability of two nodes in the obtained sequence.

We learn the relationship between items and select the user's HP_l recent behaviors to complete the screening through the similarity between items. Therefore, we will not get too large values when dealing with long sequences. When dealing with short sequences, it will not get too small values. We can deal with the problem of non-duplicate items, which leads to the problem of excessive deviation of N estimation when the sequence is too long. However, the traditional method will show obvious variation in dealing with this scene. M_s will get too large N. The N calculated by M_r is 1, and the value it gets will be too small.

Table 2 Hyperparameters under five datasets

Dataset	NOWP	RETLR	RSC15	CLEF	TMALL
HP_l	10	10	10	10	10
HP_r	0.96	0.86	0.77	0.90	0.73
HP_s	0.65	0.8	0.59	0.67	0.56

4.2 Details of Method M_{bpaa}

Method motivation: Our task is to predict the size of the user's next interactions based on history. This task can also be understood as *how large the set of candidates can determine the user's next interaction given the historical data.* We need to find the value of N when Top-N accuracy is high. As N increases, Top-N accuracy will keep increasing, so we are not finding the N that makes Top-N take the maximum. When N is infinite, Top-N accuracy is maximum and equal to 1. The user's behavior is divided into regular and irregular, and we are trying to count the N of the regular part. If we counted the irregular part of mobile behavior, we would have to rely on all the places in the world because it is possible that the user suddenly went to a new location at one time. But we know that's not appropriate. We need to find the N that does have a significant effect on the accuracy improvement. After all, randomly adding some random behavior will also make the accuracy slowly increase. Our goal is to find the moment when the Top-N accuracy does not improve significantly with the increase of N.

Our method M_{lc} is a simple recommendation model, and we can find its one-to-one correspondence with the general recommendation model. The specific correspondence is shown in Table 3. The general procedure for sequential recommendation is as follows. Firstly, we intercept the last L behaviors and input them into the recommendation model for learning. Secondly, the model predicts the global item probability for the user to be predicted and calculates the probability of items that the user interacts with next. Finally, the model recommends the Top-N items with the highest probability to the user to observe whether the user interacts with them. We can observe that L corresponds to HP_l, indicating how long the behavior is used to learn the model. HP_s corresponds to the probability calculated by the model, indicating the degree to which the candidate item conforms to the model. The accuracy corresponds to HP_r. The most important is that the N of Top-N corresponds to the N that we finally calculated. The N in Top-N means that the current model is used to screen out the N items that best meet the requirements to predict the next behavior. The final N calculated by M_{lc} is the number of nodes that meet the condition after being screened by M_{lc}.

Therefore, the operation steps of our proposed M_{bpaa} (Algorithm 2) are as follows:

M_{bpaa} is a further development of M_{lc}. M_{bpaa} uses 10 methods to find the best model to distinguish positive and negative samples most accurately. We thus get a general idea of the best performance of the recommendation algorithm in

Table 3 The relationship between M_{lc} and M_{bpaa}

Method	M_{lc}	M_{bpaa}
Corresponding parameters	HP_l	Length of user behavior
	HP_r	Accuracy
	HP_s	Predicted probability
	N	*Top-N*

Algorithm 2 Method best prediction algorithm available

```
Require: 10 recommendation algorithms
Ensure: Candidate set size $N$
Find 10 excellent methods among sequential recommendations.
Use 10 methods to calculate their respective accuracy in the case of $Top$-$N$ recommendation for different datasets. $N$ takes a positive integer from 1 to 400.
The corresponding abscissa $N$ results when the optimal accuracy growth rate slows down.
```

terms of accuracy on these datasets. Obviously, the accuracy of *Top-N* recommendation will continue to increase as N increases. Even if the model only adds some random samples, the accuracy will increase. Therefore, when the optimal accuracy growth rate slows down, the corresponding abscissa is N. With enough time to implement method M_{bpaa}, we can get a more accurate N. If we want to get N quickly, we can use method M_{lc}, which already has a good performance.

Table 4 shows a comparison of our proposed methods M_{lc} and M_{bpaa} with existing works M_s and M_r. M_s and M_r are generally suitable for cases where user behavior tends to stabilize over time and the behaviors do not keep growing. The advantage is that the methods are simple and fast. Typical scenarios are user movement behavior and calling behavior. M_{lc} and M_{bpaa} are suitable for cases where the set of behaviors is huge, and the set of user behaviors is increasing. Typical scenarios are user shopping, watching movies, and listening to music. In these scenarios, M_s and M_r will have large deviations. M_{lc} and M_{bpaa} have different characteristics from each other. M_{lc} is faster, and M_{bpaa} is more accurate.

5 Experiment

5.1 Datasets

We have done experimental verification on 5 real-world public datasets. The dataset covers the three major areas of e-commerce, music, and news. We have made basic statistics on the basic information of the dataset, such as sessions and items. The specific values are shown in Table 5.

- NOWPLAYING: The NOWP dataset was created based on music-related tweets in which users posted the tracks they are currently listening to.
- RetailRocket: RetailRocket, an e-commerce personalization company, released this dataset, which covers six months of user browsing activities.
- RSC15: It was released in the context of the ACM RecSys 2015 Challenge, which contains the recorded click sequence (item view, purchase) for six months.
- CLEF: This dataset has been provided to participants of the 2017 CLEF NewsREEL Challenge. It consists of user actions and article publishing events collected by plista for multiple publishers.
- TMall: This dataset was released in the context of the TMall competition, which contains one year's interaction log of the tmall.com website.

5.2 Analysis of behavioral similarity

After getting the embedding of the item through word2vec, we use the cosine similarity of the embedding to calculate the degree of association between the two items. The results obtained on the 5 datasets are shown in Fig. 2. The abscissa represents the distance between two items, and the distance between adjacent rows is 1. The ordinate represents the cosine similarity of embedding. The larger the value, the higher the similarity between the two items. However, the results on the 5 datasets are different. But they have some common characteristics:

- The dependency between two items decreases as their interval time increases. This shows that most of the influence on users' future behavior comes from recent events.
- As the distance increases, there is a threshold beyond which the dependence between items decreases sharply until the similarity between items resembles the similarity between random behaviors.

HP_l determines how long the user's historical data is taken to filter the next-moment items. The filtering method is to find those items that are more similar to HP_r in HP_l items than HP_s in the global items. Too long HP_l will consider those items that are not very relevant to the current item. Ultimately,

Table 4 The comparison of our proposed methods M_{lc} and M_{bpaa} with existing methods M_s and M_r. We illustrate the advantages of each method and introduce the characteristics of the usage scenarios and typical scenarios

Method	Advantages	Characteristics of usage scenarios	Example scenarios
M_s	Simple	Stable set of user behaviors	Movement behavior
M_r	Fast	New behaviors rarely appear	Calling behavior
M_{lc}	Balancing usability and accuracy	Huge set of user behaviors	Shopping
M_{bpaa}	Accurate	Massive new behaviors constantly appear	Watching movies

Table 5 The detailed statistical characteristics of the five datasets

Dataset	Timespan in days	Items	Sessions	Actions
NOWPLAYING	530	57,161	33,730	637,143
RetailRocket	133	72,076	55,351	506,875
RSC15	182	27,042	78,180	2,302,985
CLEF	28	1,290	166,854	2,154,169
TMall	90	8,854	64,262	1,029,458

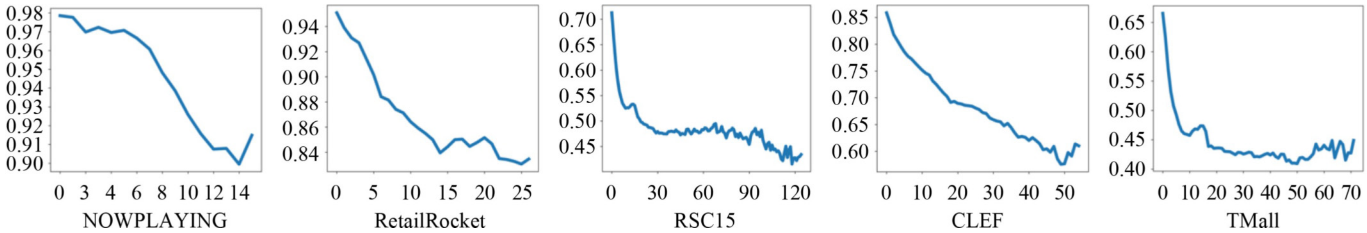

Fig. 2 The similarity between items varies with the distance

this will make HP_r and HP_s smaller. The filtering criteria will be less stringent, thus allowing those irrelevant items to satisfy the conditions. A too-short HP_l also reduces the filtering stringency, each historical item is equivalent to one filtering condition, and a shorter HP_l makes less restrictive conditions. So HP_l cannot be too short, while the historical items at that length remain highly similar to the current items, thus ensuring that HP_r and HP_s cannot be too small.

Figure 2 shows the variation of item similarity with the distance. The figure shows that the similarity is high for distances less than 10, and will be clearly distinguished from the similarity at long distances. Therefore, under the guidance of HP_l requirement, we set HP_l to 10. Although the mutation of similarity is perhaps more obvious under different datasets when HP_l is other values, it is a common pattern that the similarity changes abruptly at HP_l. The experimental results show that the distinction between different datasets is not particularly obvious. Therefore, we chose a common value that can reduce the workload and improve the method's usability.

5.3 Distribution of behavioral similarity

To filter and limit the following behavior, we selected the user's most recent 10 behaviors. We found the HP_r and HP_s by further analyzing the correlation between the behaviors. As shown in Fig. 3, we visualize the similarity between the last 10 behaviors of the user and the similarity between random behaviors. The most recent 10 behaviors are significantly higher than the similarity between random behaviors. There is a clear difference between the two distributions. But how much similarity is chosen to separate the two on both NOWPLAYING and RetailRocket datasets is not easy to determine directly. To enhance the operability of the method, HP_s calculation method is: to find historical behavior and random behavior similarity distribution peak corresponding to the abscissa x_p, x_n.

$$HP_s = \frac{(x_p + x_n)}{2}. \tag{8}$$

By setting the HP_s to the mean of x_p and x_n, a reasonable threshold can be selected to distinguish between real historical behavior and random behavior. After obtaining the HP_s, we further count the proportion of the real historical behavior that is greater than the threshold to obtain the HP_r. Through the above operations, we can obtain the hyperparameters in M_{lc} and the conditions for screening the candidate set. The specific experimental results are shown in Table 2. After obtaining the three hyperparameters in M_{lc}, we can calculate N. For the 5 datasets, we can calculate N as shown in Table 6.

From [5], we know that N significantly impacts predictability when the predictability is less than 90% and N is less than 50. After N is greater than 50, increasing N will still impact predictability, but the degree of impact will become smaller and smaller. Table 6 shows that the N_{lc} is not as accurate as N_{bpaa}. But they're pretty close. And both N_s and N_r are small (< 50). Suppose the theoretical N is 100, $N_{lc} = 160$, and $N_s = 40$. The absolute errors $|160-100| = |100-40|$ are equal. But the predictability obtained based on $N = 160$ is more accurate than that based on $N = 40$. N_{lc} is not as fragile as N_s and N_r.

One goal of the M_{lc} is to find the exact N, and another is to make it easier for everyone to use. More experiments can be added to find the optimal value when choosing HP_l, HP_r and HP_s, but this will significantly increase the workload of

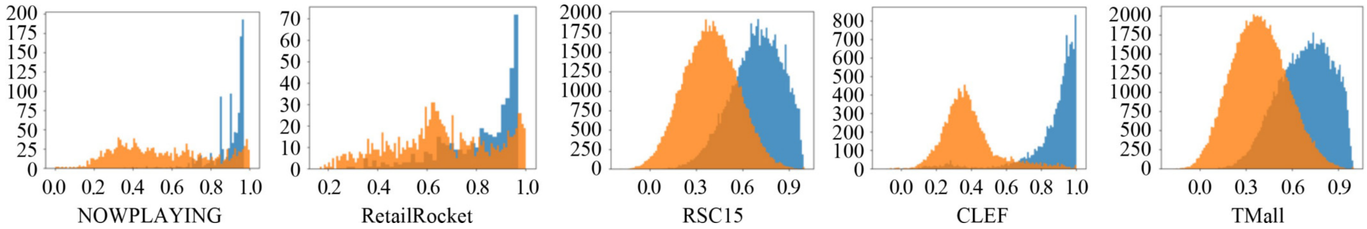

Fig. 3 Similarity distribution between historical behaviors and similarity distribution between random behaviors

Table 6 N calculated under five datasets

Dataset	NOWPLAYING	RetailRocket	RSC15	CLEF	TMall
N_s	9	4	20	22	17
N_r	1	2	1	2	1
N_{lc}	139	64	74	48	45
N_{bpaa}	100	97	80	53	70

everyone using it. The improvement in precision achieved is not as much as seen in the previous paragraph's analysis. Therefore, we adopt the parameter selection method in this paper to ensure accuracy and usability.

5.4 *Top-N* results of the 10 recommendation algorithms

We have selected 10 methods for sequential recommendation. We started by selecting several rule mining algorithms for time series, such as Simple Association Rules (AR), Markov Chains (MC), and Sequential Rules (SR). Statistical learning as an essential tool for data mining based on Bayesian theory is widely used in recommendations, so we also chose the Bayesian Personalized Ranking (BPR). Despite the simplicity of the method, neighbor-based methods often have incredible performance, so we decided on the Item-based KNN (IKNN). Matrix decomposition is a significant class of methods for recommendations. We selected Factorized Personalized Markov Chains (FPMC), Factored Item Similarity Models (FISM), Factorized Sequential Prediction with Item Similarity Models (FOSSIL), and Session-based Matrix Factorization (SMF). Gru4Rec was also added to our collection as a representative algorithm for dealing with sequence prediction in deep learning. There is a more detailed introduction to them in this paper [33]. Then we observe the performance results of 10 algorithms on 5 datasets in the case of *Top-N* recommendation. It can be observed from Fig. 4 that initially as the value of N continues to increase, the optimal accuracy will continue to be greatly improved. However, as the value of N continues to increase, the increase in the optimal accuracy will slow down significantly. On the 5 datasets, the inflection point will be slightly different. As shown in Fig. 4, on the NOWPLAYIING dataset, the inflection point appears when $N = 100$. From this, we can know that when N is less than 100, the increase of N is obviously useful for prediction accuracy. This just shows that this N is exactly the size of the candidate item sets. The specific results of N on the 5 datasets are shown in Table 6.

5.5 The results of the N

For different datasets, we calculated 4 N respectively. Among them, N_s and N_r correspond to the previous method M_s and method M_r respectively. And N_{lc} and N_{bpaa} respectively correspond to the two calculation theories we proposed, corresponding to Method M_{lc} and Method M_{bpaa}, respectively. From Table 6, we can observe that the value of N calculated by M_s and M_r is generally very small. This is because the theoretical upper limit of the N calculated by method M_s and M_r will not exceed the sequence length. The length of the sample sequence in the dataset is generally small. This calculation method deviates from the facts. For example, when a user buys 2 scorching products. Then methods M_s and M_r will consider that the number of places visited next by the user does not exceed 2. The actual situation is we only have so few user behaviors, and he interacts with hot products. The products he may interact with next should be more difficult to determine, and the number of products he may interact with should be relatively large. In the sequential recommendation, due to the defects of methods M_s and M_r itself, it cannot adapt to the calculation of N in this scenario. Our method can calculate more reasonable values. M_{bpaa} is a further optimization of M_{lc}. Although M_{bpaa} can obtain a more accurate value of N than M_{lc}, the implementation of M_{lc} is simpler and less time-consuming.

5.6 The results of the predictability

The entropy of the user sequence is calculated using the Eq. (3), and the corresponding N is calculated using four methods. We can use the Eq. (4) to calculate the related predictability. The result is shown in Fig. 5. Each dataset corresponds to 5 accuracies, corresponding to the five bars in the figure. The first bar represents the accuracy of the current optimal algorithm in $Top\text{-}N_{bpaa}$. The second and third bars respectively represent the predictability obtained by using the M_s and M_r. The fourth and fifth bars represent the predictability calculated by M_{lc} and M_{bpaa}, respectively. It can be seen from the figure that due to the shortcomings of the M_s and M_r, when quantifying the recommendation dataset, the calculated candidate set size is too small, resulting in little predictability. Since the recommendation system rarely has duplicate items when the behavior sequence is not particularly long, the existing method M_r will be severely biased and keep getting very small N. Eventually, it will make the predictability derived based on this method significantly lower than $Top\text{-}N_{bpaa}$ all the time. Correctly quantifying the size of the candidate set is extremely important for calculating predictability. The final result shows that the predictability of sequential recommendation under the five datasets is between 64% and 80%.

6 Discussion

Research on recommender systems is in full swing. However, researchers are wondering whether there is much room for recommender systems to grow. The RecSys 2019 Best Paper [34] points out that there has not been much progress in

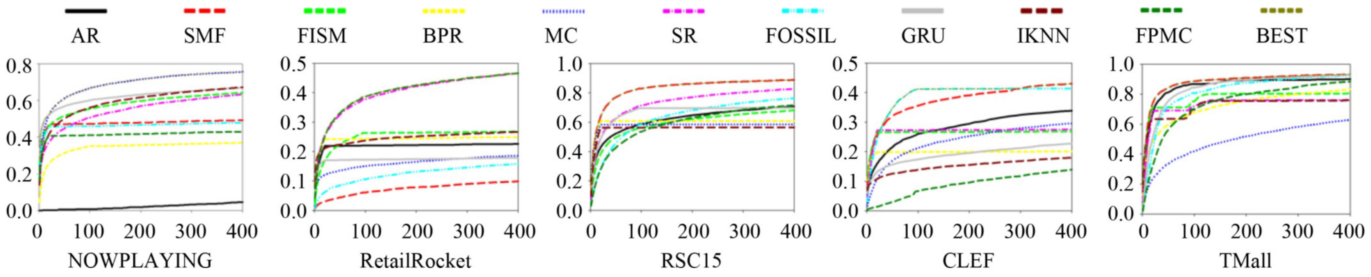

Fig. 4 Accuracy results of 10 algorithms on 5 datasets. The horizontal coordinate represents the candidate set size N. The vertical coordinate represents the accuracy corresponding to *Top-N*

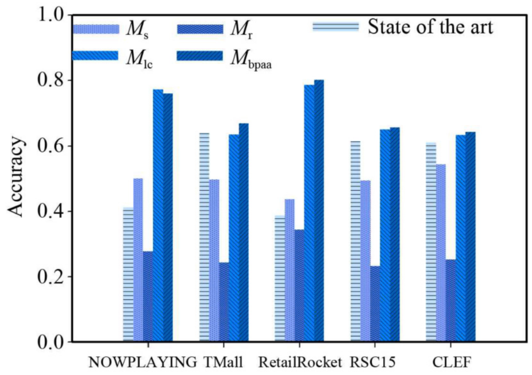

Fig. 5 The Top-N_{bpaa} accuracy of the optimal algorithm and the predictability calculated by the four methods

recommender systems recently. This is also a concern that the accuracy improvement in sequential recommendation has not been apparent. By understanding the gap between the accuracy of the current approach and predictability, we can better understand how much effort is required and how much improvement can be achieved. Knowing the predictability in different scenarios gives us a research direction and a better way to quantify the difficulty of improving the accuracy. However, the accuracy of sequential recommendation can also be enhanced through a number of behavior sequences and characteristics due to the increasingly large and diversified user data that can be recorded today. This is also the direction of recommender system development. Research on the predictability of multi-source sequence data is extremely scarce, which is also a problem that we need to solve in the future.

The user candidate set's exact size is unknown, not even by the user himself. The existing methods for measuring the N for mobile behavior are not theoretical N, but they are close to the theoretical N, so they serve well in this domain. We found that these two methods show significantly biased in the sequential recommendation. Although our approach is not guaranteed to be the theoretical N, it is close to the theoretical N and well-suited for the sequential recommendation.

7 Conclusion

To explore the predictability of sequential recommendation, we borrowed from the predictability theory of human movement behavior. The calculation of predictability involves two crucial parameters, one is the entropy S, which represents the degree of sequence confusion, and the other is the candidate set size (N). Applying the predictability theory of human movement behavior to the sequential recommendation, the existing methods can obtain accurate S, but there will be large deviations when calculating N. Inaccurate N will lead to erroneous predictability. Due to the wide range of user behavior sequence length and the increasing number of items that have never appeared before, we propose two methods to obtain a reasonable N. The results show that there is still a big gap between the accuracy of current recommendation algorithms and the predictability. At the same time, the present method introduced in this article has some limitations, which should be resolved in future work. We only considered the predictability of a single pure sequence. We should consider the impact of the user's characteristics on predictability.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 61960206008, 62002294), and the National Science Fund for Distinguished Young Scholars (61725205).

References

1. Wang S, Hu L, Wang Y, Cao L, Sheng Q Z, Orgun M. Sequential recommender systems: challenges, progress and prospects. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. 2019, 6332–6338
2. Hidasi B, Karatzoglou A, Baltrunas L, Tikk D. Session-based recommendations with recurrent neural networks. In: Proceedings of the 4th International Conference on Learning Representations. 2016
3. Li Z, Zhao H, Liu Q, Huang Z, Mei T, Chen E. Learning from history and present: next-item recommendation via discriminatively exploiting user behaviors. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018, 1734–1743
4. Song C, Qu Z, Blumm N, Barabási A L. Limits of predictability in human mobility. Science, 2010, 327(5968): 1018–1021
5. Smith G, Wieser R, Goulding J, Barrack D. A refined limit on the predictability of human mobility. In: Proceedings of 2014 IEEE International Conference on Pervasive Computing and Communications. 2014, 88–94
6. Yap G E, Li X L, Yu P S. Effective next-items recommendation via personalized sequential pattern mining. In: Proceedings of the 17th International Conference on Database Systems for Advanced Applications. 2012, 48–64
7. Ren S, Guo B, Li K, Wang Q, Yu Z, Cao L. CoupledMUTS: coupled multivariate utility time series representation and prediction. IEEE Internet of Things Journal, 2022, doi: 10.1109/JIOT.2022.3185010
8. Garcin F, Dimitrakakis C, Faltings B. Personalized news recommendation with context trees. In: Proceedings of the 7th ACM Conference on Recommender Systems. 2013, 105–112
9. Wu C Y, Ahmed A, Beutel A, Smola A J, Jing H. Recurrent recommender networks. In: Proceedings of the 10th ACM International Conference on Web Search and Data Mining. 2017, 495–503
10. Xu E, Yu Z, Guo B, Cui H. Core interest network for click-through rate prediction. ACM Transactions on Knowledge Discovery from Data, 2021, 15(2): 23
11. Tang J, Wang K. Personalized Top-N sequential recommendation via convolutional sequence embedding. In: Proceedings of the 11th ACM International Conference on Web Search and Data Mining. 2018, 565–573
12. Wu S, Tang Y, Zhu Y, Wang L, Xie X, Tan T. Session-based recommendation with graph neural networks. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence. 2019, 346–353
13. Takaguchi T, Nakamura M, Sato N, Yano K, Masuda N. Predictability of conversation partners. Physical Review X, 2011, 1(1): 011008
14. Baumann P, Santini S. On the use of instantaneous entropy to measure the momentary predictability of human mobility. In: Proceedings of the 14th IEEE Workshop on Signal Processing Advances in Wireless Communications. 2013, 535–539

15. McInerney J, Stein S, Rogers A, Jennings N R. Exploring periods of low predictability in daily life mobility. In: Proceedings of Mobile Data Challenge by Nokia. 2012
16. Krumme C, Llorente A, Cebrian M, Pentland A, Moro E. The predictability of consumer visitation patterns. Scientific Reports, 2013, 3: 1645
17. Nguyen T, Rokicki M. On the predictability of non-CGM diabetes data for personalized recommendation. In: Proceedings of 2018 CIKM Workshops Co-located with the 27th ACM International Conference on Information and Knowledge Management. 2018
18. Zhang P, Xue L, Zeng A. Predictability of diffusion-based recommender systems. Knowledge-Based Systems, 2019, 185: 104921
19. Järv P. Predictability limits in session-based next item recommendation. In: Proceedings of the 13th ACM Conference on Recommender Systems. 2019, 146–150
20. Ben-Naim A. Elements of information theory. In: Ben-Naim A, ed. A Farewell To Entropy: Statistical Thermodynamics Based on Information. Singapore: World Scientific, 2008
21. Kontoyiannis I, Algoet P H, Suhov Y M, Wyner A J. Nonparametric entropy estimation for stationary processes and random fields, with applications to English text. IEEE Transactions on Information Theory, 1998, 44(3): 1319–1327
22. Zhao Z D, Yang Z, Zhang Z, Zhou T, Huang Z G, Lai Y C. Emergence of scaling in human-interest dynamics. Scientific Reports, 2013, 3: 3472
23. Zhang L, Liu Y, Wu Y, Xiao J. Analysis of the origin of predictability in human communications. Physica A: Statistical Mechanics and its Applications, 2014, 393: 513–518
24. Wang J, Mao Y, Li J, Xiong Z, Wang W X. Predictability of road traffic and congestion in urban areas. PLoS One, 2015, 10(4): e0121825
25. Ren W, Li Y, Chen S, Jin D, Su L. Potential predictability of vehicles' visiting duration in different areas for large scale urban environment. In: Proceedings of 2013 IEEE Wireless Communications and Networking Conference. 2013, 1674–1678
26. Zhao K, Khryashchev D, Freire J, Silva C, Vo H. Predicting taxi demand at high spatial resolution: approaching the limit of predictability. In: Proceedings of 2016 IEEE International Conference on Big Data. 2016, 833–842
27. Li Y, Jin D, Hui P, Wang Z, Chen S. Limits of predictability for large-scale urban vehicular mobility. IEEE Transactions on Intelligent Transportation Systems, 2014, 15(6): 2671–2682
28. Xu T, Xu X, Hu Y, Li X. An entropy-based approach for evaluating travel time predictability based on vehicle trajectory data. Entropy, 2017, 19(4): 165
29. Chen Y Z, Huang Z G, Xu S, Lai Y C. Spatiotemporal patterns and predictability of cyberattacks. PLoS One, 2015, 10(5): e0124472
30. Fiedor P. Frequency effects on predictability of stock returns. In: Proceedings of 2014 IEEE Conference on Computational Intelligence for Financial Engineering & Economics. 2014, 247–254
31. Dahlem D, Maniloff D, Ratti C. Predictability bounds of electronic health records. Scientific Reports, 2015, 5: 11865
32. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. In: Proceedings of the 1st International Conference on Learning Representations. 2013
33. Ludewig M, Jannach D. Evaluation of session-based recommendation algorithms. User Modeling and User-Adapted Interaction, 2018, 28(4–5): 331–390
34. Dacrema M F, Cremonesi P, Jannach D. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In: Proceedings of the 13th ACM Conference on Recommender Systems. 2019, 101–109

En Xu received the bachelor's degree from Northwestern Polytechnical University, China. He is currently a PhD student with the School of Computer Science, Northwestern Polytechnical University, China. His research interests include recommender system and predictability.

Zhiwen Yu is currently a professor of the School of Computer Science, Northwestern Polytechnical University, China. He is the associate editor or editorial board of IEEE Transactions on Human-Machine Systems, IEEE Communications Magazine, ACM/Springer Personal and Ubiquitous Computing (PUC). His research interests include ubiquitous computing and mobile crowd sensing.

Nuo Li received the bachelor's degree from Northwestern Polytechnical University, China. At the moment, she is a PhD student with the School of Computer Science, Northwestern Polytechnical University, China. Her research interests include social and community intelligence and crowd knowledge transfer.

Helei Cui is a professor from Northwestern Polytechnical University, China. He received his PhD degree in Computer Science from City University of Hong Kong (CityU), China in October 2018, under the supervision of Prof. Cong Wang (IEEE Fellow). Before that, he obtained MSc degree in Information Engineering from The Chinese University of Hong Kong (CUHK), China in November 2013 and BEng degree in Software Engineering from Northwestern Polytechnical University, China in July 2010. His research interests include industrial internet, secure crowdsensing, and distributed storage networks.

Lina Yao is currently a scientia associate professor at School of Computer Science and Engineering, the University of New South Wales (UNSW), Australia. She received her PhD degree and Master degree both from The University of Adelaide (UoA), Australia in 2014 and 2010, respectively, and her Bachelor degree from Shandong University (SDU), China. Her research interest lies in data mining and machine learning applications with the focuses on internet of things analytics, recommender systems, human activity recognition, and brain computer interface.

Bin Guo is a professor from Northwestern Polytechnical University, China. He received his PhD degree in computer science from Keio University, Japan in 2009 and then was a post-doc researcher at Institut TELECOM SudParis in France. His research interests include ubiquitous computing and mobile crowd sensing.