

Multi-factor Sequential Re-ranking with Perception-Aware Diversification

Yue Xu

yue xu.xy@foxmail.com
Alibaba Group.

Hao Chen

sundaychenhao@gmail.com
The Hong Kong
Polytechnic University,
Hong Kong.

Zefan Wang

wongzfn@gmail.com
Jinan University, China.

Jianwen Yin

yjw264077@alibaba-
inc.com
Alibaba Group.

Qijie Shen

qijie.sqj@alibaba-inc.com
Alibaba Group.

Dimin Wang

dimin.wdm@alibaba-
inc.com
Alibaba Group.

Feiran Huang

huangfr@jnu.edu.cn
Jinan University, China.

Lixiang Lai

lixiang.llx@alibaba-
inc.com
Alibaba Group.

Tao Zhuang

zhuangtao.zt@alibaba-
inc.com
Alibaba Group.

Junfeng Ge

beili.gjf@alibaba-inc.com
Alibaba Group.

Xia Hu

xia.hu@rice.edu
Rice University, USA

ABSTRACT

Feed recommendation systems, which recommend a sequence of items for users to browse and interact with, have gained significant popularity in practical applications. In feed products, users tend to browse a large number of items in succession, so the previously viewed items have a significant impact on users' behavior towards the following items. Therefore, traditional methods that mainly focus on improving the accuracy of recommended items are sub-optimal for feed recommendations because they may recommend highly similar items. For feed recommendation, it is crucial to consider both the accuracy and diversity of the recommended item sequences in order to satisfy users' evolving interest when consecutively viewing items. To this end, this work proposes a general re-ranking framework named Multi-factor Sequential Re-ranking with Perception-Aware Diversification (MPAD) to jointly optimize accuracy and diversity for feed recommendation in a sequential manner. Specifically, MPAD first extracts users' different scales of interests from their behavior sequences through graph clustering-based aggregations. Then, MPAD proposes two sub-models to respectively evaluate the accuracy and diversity of a given item by capturing users' evolving interest due to the ever-changing context and users' personal perception of diversity from an item sequence perspective. This is consistent with the browsing nature of the feed scenario. Finally, MPAD generates the return list by sequentially selecting optimal items from the candidate set to maximize the joint benefits of accuracy and diversity of the entire list. MPAD has been implemented in Taobao's homepage feed to serve the main traffic

and provide services to recommend billions of items to hundreds of millions of users every day.

KEYWORDS

feed recommendation, diversified recommendation, re-ranking

1 INTRODUCTION

Feed recommendation system (RS) is one type of product that recommends a sequence of items for users to browse and interact with. It has been widely applied to various online platforms, such as on the homepage of Kuaishou [27], Xiaohongshu [21], Taobao [35], and AliExpress [18]. An example of the feed recommendation is given in Figure 1. The feed allows users to continuously scroll down the item list of items in the viewing window, such that previously viewed items have a large impact on users' behaviors towards the next item. In this case, traditional methods that mainly focus on improving the accuracy of recommended items become sub-optimal for feed recommendation because they usually ignore the correlations between consecutive items. For example, if a user was shown a mobile phone item, it may be sub-optimal to put a series of more mobile phone items next to it. This mismatch is exacerbated by the fact that similar items tend to have similar click ratios. Therefore, it is of vital importance for feed recommendation methods to consider both accuracy and diversity from an item sequence perspective to attract users to browse and interact with more items in the feed [21, 27, 35].

This work focuses on two common characteristics that are closely related to optimizing accuracy and diversity in feed recommendations. First, different users may have different perceptions of diversity, such that item diversity in feed recommendations should be measured based on their personal interests. Existing works on diversified recommendations mainly focus on measuring the dissimilarity between item pairs, without considering users' personal interests. For example, the post-processing methods improve diversity by heuristically rearranging the item order based on predefined

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD 2023, August 6, 2023, Long Beach, California, United States

© 2023 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>



Figure 1: An example of different forms of real-world RS. Left: Traditional RS recommends items in different channel blocks. Right: Feed RS recommends a sequence of items where users can slide down to view more items.

rules, which are not customized for all users [3, 6–8, 37, 41]; the learning-based methods measure the similarity of a given item pair by directly comparing the item embeddings [1, 9, 44, 46]. Though effective, the ignorance of users’ personal interests may lead to a mismatch between the model’s definition of diversity and users’ perceptions of diversity. For example, some female customers prefer to view more clothing than others in the feed. Directly reducing the probability of presenting user-preferred items to increase diversity may degrade their satisfaction with the recommended results.

Second, in feed applications, users tend to view many items in a row, such that users’ interests may evolve during this continuous browsing process. In this case, the measurement of both item accuracy and diversity should consider the evolving interest due to the ever-changing context so as to accommodate the sequential browsing nature in feed scenarios. However, most existing interest models mainly focus on learning users’ interests from their historical behaviors with less emphasis on the evolution of interests along with the browsing context [10, 34, 47, 48]. Another line of research on re-ranking proposes various list-wise solutions to capture the interior correlations between items within the context [2, 5, 20, 32]. Nevertheless, they mainly focus on improving accuracy regardless of diversity. Some recent works devote efforts to solve this accuracy-diversity dilemma and obtained promising results [21, 27, 44, 46]. However, the joint optimization of accuracy and diversity still remains to be a challenging problem, especially for industrial implementation on large-scale systems.

In light of the above challenges, in this paper, we investigate the following research questions. 1) How to formulate and design a general framework for jointly optimizing accuracy and diversity from an item sequence perspective? 2) How to estimate accuracy and diversity with adaptation to the evolution of user interest while browsing consecutive items in feed scenarios? 3) How to implement the proposed framework in industrial systems for practical applications and how well does it perform? To this end, we propose a general Multi-factor Sequential Re-ranking with Perception-Aware

Diversification (MPAD) framework to jointly optimize accuracy and diversity for practical feed recommendations. This framework consists of four main components. The bi-sequential determinantal point process (BS-DPP) algorithm provides a principled and tractable framework for sequential item selection to maximize the joint benefits of accuracy and diversity of the entire item list. The Multi-scale Interest Extraction (MIE) model extracts multi-scale user interests through graph clustering-based aggregations. The Context-aware Accuracy Estimation (CAE) model provides an estimate of context-aware accuracy from a sequence perspective by learning from both the multi-scale interests and the ever-changing browsing context. The Perception-Aware Kernel (PDK) evaluates the similarity between items with consideration of the user’s perception of diversity based on personal interests. The main contributions are as follows.

- This work formulates the feed recommendation task as a multi-factor re-ranking problem and proposes a principle and tractable MPAD framework to maximize the joint benefits of accuracy and diversity of the entire recommended item list.
- This work proposes a series of collaborative models to estimate the accuracy and diversity of an item list from a sequence perspective. They are able to capture the influence from both the browsing context and the evolving user interests to align with the browsing nature of the feed scenario. We also propose a tailored BS-DPP algorithm to jointly optimize the accuracy and diversity when selecting optimal items in a sequential manner.
- This paper presents a general system architecture for the deployment of MPAD in industrial systems. It has now been implemented in the homepage feed to achieve 2.4% lift on user clicks, 2.0% lift on stay time, and 4.0% lift on content diversity. The architecture now serves Taobao’s main traffic with 120,000 queries-per-second at peak.
- This work conducts extensive experiments on both offline datasets and online A/B tests. The results show that our proposed MPAD significantly outperforms other methods. The source code has been made public¹.

2 PROBLEM SETUP

A typical pipeline of industrial RS includes three stages [21, 44], i.e., matching, ranking, and re-ranking. The RS first retrieves candidate items from item databases at the matching stage. Then, the ranking modules measure item accuracy in a point-wise manner. Finally, the top items will be sent to the re-ranking module to determine the final item list to present to users.

In this paper, we consider a multi-factor feed recommendation problem at the re-ranking stage, where the task is to select an item sequence $S = \{i_1, i_2, \dots, i_K\}$ with size K from a candidate set $I = \{i_1, i_2, \dots, i_N\}$ with size $N \gg K$ provided by the ranking module. The selection of sequence S depends on both the *item accuracy* which relates to user’s preference for the items and the list-wise *item diversity* which influences user’s intention to browse and interact in practical applications [12, 21, 29, 44]. Formally, given a target user u and a set of candidate items I , our aim is to select a fixed-size subset from I and determine their order in a page to

¹The source code is available at <https://anonymous.4open.science/r/MPAD/>.

maximize a joint utility function:

$$\mathcal{P}_0 : \arg \max_{S \subseteq I} f(u, S) = F(\text{Acc}(u, S), \text{Div}(u, S)), \quad (1)$$

where the first term $\text{Acc}(u, S)$ evaluates the context-aware item accuracy based on user interests and browsing context, the second term $\text{Div}(u, S)$ evaluates the list-wise diversity of all items, and the fusion function $F(\cdot)$ measures the contribution of item accuracy and diversity to the joint utility $f(u, S)$.

Note that this formulation extends the commonly used item-level diversity [9, 21, 27, 44] to personalized user-item-level diversity, i.e., evolving from $\text{Div}(S)$ to $\text{Div}(u, S)$. As such, the solution needs to consider user's *personalized* perception of diversity on the recommended results.

3 METHODOLOGY

This section first gives an overview of the proposed MPAD framework in Sec. 3.1. Then, this section introduces the main building blocks of MPAD in order from Sec. 3.2 to Sec. 3.5. Finally, this section discusses the online implementation of MPAD in Sec. 3.6.

3.1 Framework Overview

The framework consists of two layers: the *selection-layer* uses a sequential item selection algorithm to select items from the candidate set using the item accuracy and diversity scores evaluated by the *estimation-layer*. The detailed workflow is presented in Figure 2.

Specifically, the selection-layer operates with the BS-DPP algorithm which considers both list-wise item diversity and context-aware item accuracy during the selection of optimal items. It indeed offers a principle and tractable solution for function $F(\cdot)$. The estimation-layer, on the other hand, consists of three components. The *first* component is MIE which groups users and items into different clusters and represents the user's multi-scale interest based on behavior sequences encoded by item/cluster embeddings. MIE can be computed offline for online complexity reduction. The *second* component is CAE which refines the point-wise accuracy scores from the ranking stage into context-aware accuracy scores by making use of both browsing context and multi-scale user interests. The refined scores are used in the computation of $\text{Acc}(u, S)$ in BS-DPP. The *third* component is PDK which computes item similarities based on both item embedding and the user's different scales of interests. The diversity kernel measures the diversity term $\text{Div}(u, S)$ in BS-DPP.

3.2 Bi-Sequential Item Selection

This section presents the BS-DPP algorithm for incremental item selection at the selection-layer. In BS-DPP, both the item diversity scores and the item accuracy scores are considered to be sequentially updated along with the item selection process.

This is different with standard Determinantal Point Process (DPP) methods [9, 44] where the item accuracy scores are considered to be fixed values, regardless of the change of context. Therefore, BS-DPP is more in accordance with the browsing nature of feed products where the user's interests may evolve during reviewing consecutive items.

3.2.1 Task Formulation. A point process P defined on an item set $I = \{i_1, i_2, \dots, i_N\}$ is a probability distribution on the powerset of I (i.e., the set of all subsets of I), where the probability satisfies $\sum_{S \subseteq I} P(S) = 1$. The probability of choosing a specific item subset is determined by the kernel function in DPP and the item selection process is usually modeled as a MAP inference [9, 25]. In this paper, we define the DPP kernel based on a combined measurement of $\text{Acc}(u, S)$ and $\text{Div}(u, S)$, such that the probability of choosing an item subset is naturally proportional to the joint optimization of item accuracy and diversity. Based on the DPP theory [9, 25], the objective in (1) equals to:

$$\mathcal{P}_1 : \arg \max_{S \in I} F(\text{Acc}(u, S), \text{Div}(u, S)) = \log \det(K_S^u), \quad (2)$$

where K_S^u is the kernel function defined with $\text{Acc}(u, S)$ and $\text{Div}(u, S)$, to be discussed later; $\log \det(K_S^u)$ is the log-probability function of choosing a subset S for user u . In this way, the aim to maximize the utility function $f(u, S)$ in (1) is transformed into maximizing the log-probability function $h(u, S) = \log \det(K_S^u)$.

3.2.2 Bi-Sequential DPP. Standard DPP methods [9, 44] construct the kernel matrix as follows:

$$K_S^u(i, j) = g(u, i) \cdot D(i, j) \cdot g(u, j), \quad (3)$$

where $g(u, i)$ is the point-wise accuracy score evaluated between user u and item $i \in S$, regardless of the page-context, while $D(i, j)$ measures the similarity between item i and item j with $\forall i, j \in S$, regardless of user's personal interests. In contrast, BS-DPP considers that 1) the accuracy scores are related to the browsing context, i.e., the previously added items in S ; and 2) the diversity scores are related to the user's interests. This changes the definition in (3) into

$$K_S^u(i, j) = g(u, i|S) \cdot D(i, j|E_u) \cdot g(u, j|S), \quad (4)$$

where $g(u, i|S)$ denotes the context-aware accuracy score which conditions on the previously presented items in S , while $D(i, j|E_u)$ measures the similarity between item $i, j \in S$ condition on the user's interest E_u . We modify the log-probability of choosing a subset S as

$$h(u, S) = \sum_{i \in S} g(u, i|S) + \alpha \cdot \log \det(D_S^u), \quad (5)$$

where α is a tunable parameter to control the trade-off between the diversity and accuracy of the recommended results. It is useful in practical feed applications since different platforms need such a parameter to control the tendency towards accuracy or diversity to suit different business orientations, e.g., more accuracy for relevant recommendations or more diversity for discovering new interests.

The objective in (2) can be solved based on the popular greedy approximation methods [9, 16, 44], which maximize the marginal gain when incrementally adding a new item to set S . Combining with our definition of the kernel function, the greedy maximization step to choose an optimal item per iteration can be written as

$$j = \arg \max_{i \in I \setminus S} \log \det(K_{S \cup \{i\}}^u) - \log \det(K_S^u) \quad (6a)$$

$$= \arg \max_{i \in I \setminus S} h(u, S \cup \{i\}) - h(u, S) \quad (6b)$$

$$= \arg \max_{i \in I \setminus S} g(u, i|S) + \alpha \cdot (\log \det(D_{S \cup \{i\}}^u) - \log \det(D_S^u)) \quad (6c)$$

$$= \arg \max_{i \in I \setminus S} g(u, i|S) + \alpha \cdot \log(d_i^2). \quad (6d)$$

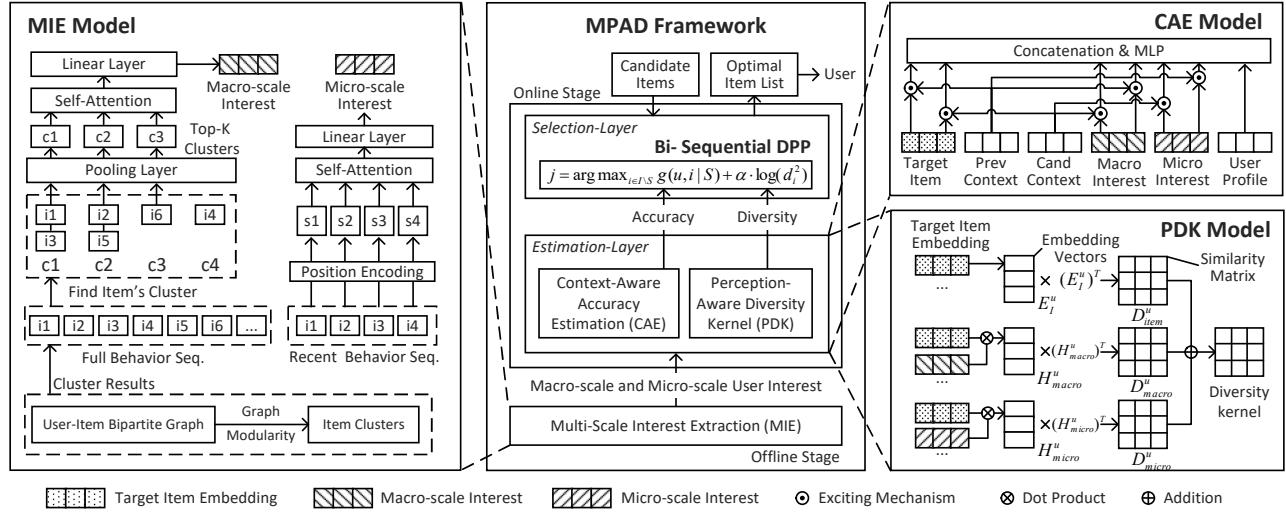


Figure 2: An overview of the MPAD framework.

Algorithm 1 Bi-Sequential Item Selection in MPAD

```

1: Initialization:
2:  $D_S, \epsilon, c_i = [], d_i^2 = D_{ii}, j = \arg \max_{i \in I} \log(d_i^2), S = \{j\}$ .
3: Iteration:
4: while  $|S| < k$  and  $d_i^2 < \epsilon$  do
5:   for  $i \in I \setminus S$  do
6:      $e_i = (D_{ji} - \langle c_j, c_i \rangle) / d_j$ .
7:     Update  $d_i^2 = d_i^2 - e_i^2, c_i = [c_i \quad e_i]$ .
8:     Update  $g(u, i|S)$  with the proposed preference model.
9:   end for
10:  Obtain  $j = \arg \max_{i \in I \setminus S} g(u, i|S) + \alpha \cdot \log(d_i^2)$ .
11:  Update subset  $S = S \cup \{j\}$ .
12: end while

```

The complete algorithm of BS-DPP in MPAD is given in Algorithm 1. We defer more details on the derivations of (6) and the update of term $\log(d_i^2)$, i.e., Step. 6 and Step. 7 in Algorithm 1, to the appendix. We now introduce how to obtain $g(u, i|S)$ and $\log(d_i^2)$ in (6d) via the proposed CAE and PDK model in the sequel.

3.3 Multi-Scale Interest Extraction

In this section, we propose the MIE model to extract users' multi-scale interests, which are used as the input of the subsequent CAE and PDK models. Existing user interest models usually directly perform self-attention on user's behavior items [10, 33, 34, 48]. However, directly mixing the information from a large quantity of raw item-level features may introduce redundant or noisy information to the model thus affecting learning performance. It is also hard for them to distinguish users' different aspects of interests, especially from the full behavior sequences.

Therefore, this work proposes MIE to describe user interests from two scales, i.e., the micro-scale and the macro-scale. The micro-scale captures users' recent interests, such as their recent attention to gold necklaces and earrings. The macro-scale, on the other hand, models

the user's long-term interests at a broader scope, such as fashion, clothing, or sports. For macro-scale interests, MIE groups items into clusters based on graph modularity and represents the user's macro-level interest through cluster-wise aggregated embeddings. Each cluster corresponds to one interest point at the macro level. For micro-scale interests, MIE directly uses item-level features of each item within the user's behavior sequence, such as item id sequences and feature sequences, for micro-scale interest modeling. Each item corresponds to one interest point at the micro-level. MIE also adopts time decay encoding to distinguish the freshness of recent micro-level interests.

Graph Clustering with Modularity. The user-item interaction data can be represented as a bipartite network. The edges (i.e., interactions) within the bipartite network only exist between user nodes and item nodes. In this paper, we partition the clusters in a user-item bipartite network based on the bipartite modularity [4], which is defined as

$$Q = \frac{1}{E} \sum_{i,j} (A_{ij} - P_{ij}) \delta(c_i, c_j), \quad (7)$$

where E is the total number of edges in the bipartite graph, A_{ij} is the adjacency matrix where the element equals one if an interaction between i and j exists, P_{ij} refers to the expected edge between i and j in a graph partitioned by different clusters, and $\delta(c_i, c_j)$ is the indicator function which equals one if i and j belongs to the same cluster, otherwise zero. A larger value of Q means that there are more edges in clusters than expected, which implies a stronger cluster structure. The graph modularity Q can be optimized in an iterative manner according to the Louvain algorithm [13]. After the algorithm converges, the items are grouped into different clusters which are used as the foundation of macro-level interest modeling.

Macro-Level User Interest. For a given user u , we first classify its behavior items into several interest points according to their belonged clusters. Each interest point represents the user's one aspect of macro-level interest. We present an example in Figure 2.

Given a user u with full behavior sequence $I^u = \{i_1, i_2, i_3, i_4, \dots, i_N\}$, we partition these behavior items into four interest points, i.e., $C^u = \{c_1, c_2, c_3, c_4\}$. Here $c_1^u = \{i_1, i_3\}$ due to that i_1 and i_3 belong to the same cluster. We obtain the representation of one interest point by pooling over the embedding of its contained items:

$$h_m^u = \text{Aggregate} \{e_{i_x}, \forall i_x \in c_m\}, \quad (8)$$

where c_m refers to the m -th interest point, e_{i_x} denotes the embedding of behavior item i_x and $\text{Aggregate}(\cdot)$ is an aggregation function, which is sum pooling in this paper.

Then, we perform multi-head attention among the top-M interest groups to obtain the representation of macro-level interests. Formally, the formulation of one single-head attention can be written as

$$\text{Att}(Q_m, K_m, V_m) = \text{Softmax} \left(\alpha Q_m K_m^T \right) V_m, \quad (9)$$

where $Q_m = h_m^u W^Q$, $K = h_m^u W^K$, and $V = h_m^u W^V$ are the linear transformations applied to the representation of interest group h_m^u . The scaling factor α is usually set to be $1/\sqrt{d}$ with d being the dimension of the embedding vector. Then, the representation of macro-level user interest via multi-head attention is

$$\text{Head}_m = \text{Att}(Q, K_m, V_m), \quad (10a)$$

$$h_{\text{macro}}^u = \text{Concat}(\text{Head}_1, \dots, \text{Head}_m) W^O, \quad (10b)$$

where $\text{Concat}(\cdot)$ denotes the concatenation of embedding vectors and W^O denotes the linear projection matrix and scales with the number of used heads.

Micro-Level User Interest. User's micro-level interests are usually more dynamic and more concrete than macro-level interests. Therefore, we directly perform multi-head attention towards the individual behavior items, instead of clusters, to obtain the representation of micro-level user interests. Noticeably, we inject the time decay corresponding to each behavior item into the embedding to describe the freshness of this aspect of interest. To be more specific, the expanded embedding of each behavior item can be written as

$$\tilde{e}_{i_x} = \text{Concat}\{e_{i_x}, t_{i_x}\}, \forall i_x \in I_{\text{micro}}^u, \quad (11)$$

where I_{micro}^u denotes the set of individual behavior items and t_{i_x} is a learnable embedding that represents the time interval from the interaction time till now. Then, we obtain the representation of the user's micro-level interest in the target item as

$$\text{Head}_i = \text{Att}(Q_i, K_i, V_i), \quad (12a)$$

$$h_{\text{micro}}^u = \text{Concat}(\text{Head}_1, \dots, \text{Head}_h) W^O, \quad (12b)$$

where Q , K , and V follows similar definition as in (9) but replace the embedding of interest group h_m^u with the embedding of each individual behavior item \tilde{e}_{i_x} .

3.4 Context-Aware Accuracy Estimation

In this section, we propose the CAE model to refine the point-wise accuracy scores produced by models in the ranking stage into context-aware accuracy scores for the measurement of $\text{Acc}(u, S)$. The proposed model only performs a linear transformation on the embedding vectors such that it is low-cost for online inference. A brief workflow of CAE is presented in Figure. 2.

CAE maintains two embeddings to describe the context information. First, when determining the k -th item in a page, we represent the context of previous reviewed items as

$$h_{\text{prev}} = \text{Aggregate}(e_i, i \in [k-1]), \quad (13)$$

where $[k-1] = \{1, 2, \dots, k-1\}$. Second, we represent the context of all candidate items to reflect the overall tendency from ranking models, which can be written as

$$h_{\text{cand}} = \text{Aggregate}(e_i, i \in [N]). \quad (14)$$

Next, we model the influence from the context of previous items and candidate items towards the target item based on an excitation mechanism proposed in SENet [19]. Taking the context of previous items as an example, we obtain its excited representation as

$$W_{\text{prev}} = \sigma(W_2 \cdot \delta(W_1 \cdot h_{\text{prev}})), \quad (15)$$

where $\sigma(\cdot)$ is the sigmoid activation function, $\delta(\cdot)$ is the Relu function, W_1 and W_2 are the linear transformation matrices. This excitation operator can be understood as a low-cost attention mechanism to extract the key information embedded in the context vector h_{prev} . Then, we multiply W_{prev} with the target item embedding to emphasize the influence from the context of previous items:

$$h_{\text{prev}}^{i_t} = W_{\text{prev}} \otimes e_{i_t}, \quad (16)$$

where \otimes denotes the dot product operation. The same goes for the excited vector for the context of candidate items. Moreover, by replacing e_{i_t} in (16) with the macro-level user interest h_{macro}^u and the micro-level user interest h_{micro}^u , we obtain another four excited vectors, i.e., $h_{\text{pr,lo}}^u$, $h_{\text{pr,sh}}^u$, $h_{\text{ca,lo}}^u$, and $h_{\text{ca,sh}}^u$, to model the drift of user interest based on the list-wise context. Finally, we concatenate all excited embeddings together and feed it into an MLP layer with softmax function to get the output scores:

$$h_{\text{all}}^u = \text{Concat}(e_{i_t}, h_{\text{prev}}^{i_t}, h_{\text{cand}}^{i_t}, h_{\text{pr,lo}}^u, h_{\text{pr,sh}}^u, h_{\text{ca,lo}}^u, h_{\text{ca,sh}}^u), \quad (17a)$$

$$\hat{Y}_{II}(u, i) = \text{Softmax} \left(\text{MLP} \left(h_{\text{all}}^u \right) \right), \quad (17b)$$

This CAE model can be trained with the commonly used cross-entropy loss as in other ranking models.

3.5 Perception-Aware Diversity Kernel

This section introduces the design of diversity kernel D_S^u in (5). In general, the diversity kernel determines how to evaluate the similarity between any given pairs of items in set S . The elements of D_S^u determines the $\log(d_i^2)$ term in (6d). Different definitions of the diversity kernel lead to disparate diversification results. In this paper, we introduce the user's multi-scale interests obtained in Sec. 3.3 into the measurement of item similarity. This connects diversity measurement with the user's personal perception of diversity due to distinct interests.

Specifically, we define an elementary kernel based on the form of SE kernel [39] for the perception on macro-level interests as

$$D_{\text{macro}}^u(i, j | E_u) = a_l^2 \exp \left[-\frac{\mathbf{h}_{i,\text{macro}}^u \otimes \mathbf{h}_{j,\text{macro}}^u}{b_l^2} \right], \quad (18)$$

where a_l^2 is the magnitude of the correlated components, b_l is its length scale, and $\mathbf{h}_{i,\text{macro}}^u$ refers to the dot product between the item embedding e_i and the macro-level interest vector $\mathbf{h}_{\text{macro}}^u$. Similar

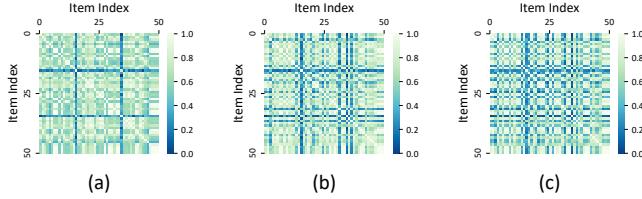


Figure 3: Influence of adding multi-scale user interests to the diversity measurement. (a) item-level similarity; (b) item-level similarity with macro-level interests; (c) item-level similarity with macro-level and micro-level interests.

goes for $h_{j,\text{macro}}^u$. The kernel for the perception on micro-level interests can be defined as

$$D_{\text{micro}}^u(i, j|E_u) = a_s^2 \exp \left[-\frac{\mathbf{h}_{i,\text{micro}}^u \otimes \mathbf{h}_{j,\text{micro}}^u}{b_s^2} \right], \quad (19)$$

where a_s^2 and b_s are hyper-parameters for micro-level interests. We also define another elementary kernel that directly compares the similarity between items based on their embeddings. In this case, the item-level diversity used in existing literature [9, 21, 27, 44] can be treated as a special case of PDK. In particular, this kernel can be defined as

$$D_{\text{item}}(i, j|E_u) = a_s^2 \exp \left[-\frac{\mathbf{e}_i^u \otimes \mathbf{e}_j^u}{b_s^2} \right], \quad (20)$$

These elementary kernels can be merged into a composite kernel without influencing the kernel properties via addition and multiplication operations [39]. More complicated operations such as automatic kernel learning are also worth trying for better adaptivity and full automation of a system, e.g., deep DPP [16], which can be explored in the future. In this work, we adopt the addition operation to construct this composite kernel:

$$D^u(i, j|E_u) = D_{\text{item}}^u(i, j|E_u) + \beta_1 \cdot D_{\text{macro}}^u(i, j|E_u) + \beta_2 \cdot D_{\text{micro}}^u(i, j|E_u), \quad (21)$$

where β_1 and β_2 are the hyper-parameters to control the influence from macro-level and micro-level interest to the diversification results.

We give an example in Figure. 3 to show the change of diversity measurement when adding user interests into the kernel. Figure. 3(a) shows the item similarity in $D_{\text{item}}^u(i, j|E_u)$ which only considering the distance of item embedding. Figure. 3(b) shows the item similarity after adding the macro-level interests $D_{\text{macro}}^u(i, j|E_u)$ into the kernel. Figure. 3(c) shows the item similarity of the complete kernel $D^u(i, j|E_u)$. It is clear that part of dissimilar items transforms into similar items due to the consideration of user interests, and vice versa. In this way, the similarity values of the same set of items are different for distinct users, thereby leading to perception-aware diversification.

3.6 Online Implementation

In this section, we introduce the online implementation of MPAD in the Homepage Feed of Taobao Mobile App. The presented system architecture is able to handle 120,000 QPS at traffic peak and respond within 20 milliseconds in general. It now serves the main

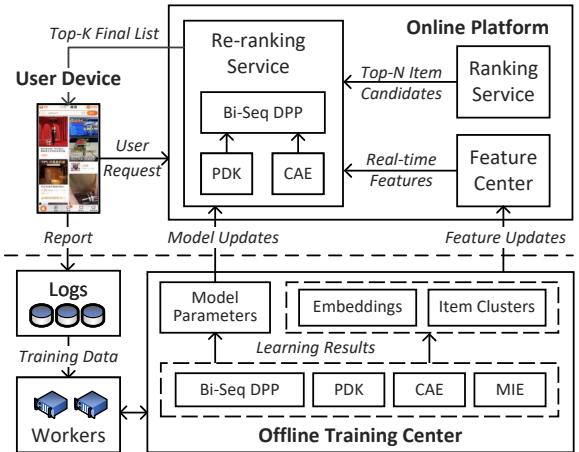


Figure 4: The System Architecture for Online Deployment

traffic of Taobao to provide services to hundreds of millions of users towards billions of items in Taobao every day.

The architecture to implement the proposed MPAD model in Taobao is presented in Fig. 4, including the workflow of both offline training and online serving. The offline training is based on a distributed machine learning platform. The learned embedding and item clustering results are uploaded to the feature center for online serving. The re-ranking service retrieves user and item features from the feature center in real-time and feeds them into a series of models to determine the final item list. Note that the graph clustering in MIE only performs offline to reduce online complexity.

The online inference complexity consists of three parts. First, each element in PDK requires the computation of dot product between two embedding vectors which incurs a complexity of $O(d)$ where d is the length of embedding, such that the overall complexity of computing PDK scales as $O(dN^2)$, where N is the number of candidate items. Note that the dot product of embedding vectors between hot items and active users can be pre-computed offline to save a lot of computations. Second, the accuracy estimation in CAE only involves linear transformation over embedding vectors in the exciting mechanism and the final MLP layer. As such, the complexity scales linearly with the length of embedding vectors, i.e., $O(dN)$ where we assume the length of each input embedding is d . Third, the BS-DPP runs in the same complexity as standard DPP [9], i.e., $O(K^3)$ time for unconstrained MAP inference and $O(K^2N)$ to return K items.

4 EXPERIMENTAL RESULTS

In this section, we conduct extensive experiments on both offline datasets and real-world online RS with the goal to answer the following research questions. **Q1:** Does MPAD outperform other SOTA methods in terms of accuracy and diversity for feed recommendation? **Q2:** How do different components of MPAD influence the final performance? **Q3:** How does MPAD perform in real-world feed recommendation platforms?

4.1 Experimental Setup

4.1.1 Datasets. We conduct offline experiments on three public available datasets: MovieLens dataset², Wechat dateset³, and Taobao dataset⁴. Specifically, MovieLens dataset is a widely-used benchmark dataset for movie recommendations, which contains 10 million samples. Here we propose it for easy reproduction. Wechat dataset is collected from 7.3 million of video playback logs on Wechat Mobile App. It is one of the largest mobile social applications in China. The dataset involves 20,000 users and 96,564 videos. The label is marked as positive if the user has watched more than 90% playback progress of a video. Taobao dataset is a widely used public benchmark dataset for online advertising, which contains over 100 million ad display/click logs collected from Taobao Mobile App. It is one of the largest online merchandise applications in China. The logs involve 1 million users and 800 thousand of ads collected on Taobao Mobile App.

4.1.2 Comparing Methods. We compare MPAD with both point-wise and list-wise mainstream methods for recommendation tasks.

Point-wise baselines: we compare with four commonly used point-wise baselines, i.e., the shallow model based on linear regression (LR) [31], the PNN model [38] which performs feature interaction with different product operations, the Wide & Deep learning model (WDL) [11] and DeepFM model [17] which adopt a hierarchical structure consists of linear and deep layers. We also compare with a few representative user interest models, i.e., DIN [48] which models short user behavior sequences with the target attention mechanism; DIEN [47] which uses an interest extraction layer based on Gated Recurrent Unit (GRU) to model users' temporal drifting interest; SIM [34] which models user's full behavior sequence based on a two-stage paradigm.

List-wise baselines: we compare with three representative list-wise baselines, i.e., DCLM [2] which applies GRU to encode the input ranking list, accompanied with a global vector to learn a powerful scoring function for list-wise re-ranking; PRM [32] which uses the self-attention mechanism to capture the mutual influence among items in the input ranking list; Seq2Slate [5] which adopts RNN and pointer network to encode the previous selected items when selecting the most appropriate item for next step. We compare with the statistical models, i.e., maximal marginal relevance (MMR) [8] and fast DPP [9]. Both of them have a tunable parameter to balance accuracy and diversity, similar to MPAD. We also compare with the generative-based models which directly generate item lists as the final results, including ListCVAE [23] and PivotCVAE[28].

4.1.3 Metrics. For accuracy estimation, we use the commonly used Area Under ROC (AUC) and Logloss (cross entropy) to evaluate the point-wise estimation performance; and use normalized discounted cumulative gain (nDCG) [22] and mean average precision (MAP) to measure the list-wise estimation performance. nDCG@K or MAP@K refers to the performance of top-k recommended items in the return list. For list-wise diversity, we use intra-list average distance (ILAD) [45] to evaluate the richness of diversified items in

Table 1: Comparison of user interest modeling (bold: best; underline: runner-up). The marker * denotes that our model performs significantly better than the runner-up with $p < 0.01$ over 25 runs.

Method	MovieLens		WeChat		Taobao	
	AUC (\uparrow)	Logloss (\downarrow)	AUC (\uparrow)	Logloss (\downarrow)	AUC (\uparrow)	Logloss (\downarrow)
LR	0.7337	0.6254	0.6433	0.6656	0.5725	0.1930
PNN	0.7836	0.5597	0.6976	0.6295	0.6309	0.1896
WDL	0.7883	0.5584	0.6968	0.6295	0.6316	0.1894
DeepFM	0.7894	0.5571	0.6979	0.6290	0.6315	0.1898
DIN	0.8024	0.5394	0.6951	0.6319	<u>0.6333</u>	0.1896
DIEN	0.8028	<u>0.5344</u>	0.6994	<u>0.6290</u>	0.6324	0.1902
SIM	0.8023	0.5349	0.7006	0.6309	0.6312	0.1893
Ours	0.8056*	0.5305*	0.7014*	0.6279*	0.6361*	0.1884*

a page. Moreover, we use PV, Stay Time, Category Breadth, CLICK, CTR, and GMV to evaluate online performance. Here, PV refers to the total number of browsed items, Stay Time is the average browsing time of all users, and Category Breadth computes the average number of distinct categories of all exposed items on all pages, reflecting the diversity of recommendation results. CLICK refers to the total number of clicked items, CTR equals CLICK/PV which measures users' willingness to click. GMV is a term used in online retailing to indicate the total sales monetary value for merchandise sold over a certain period of time. We use the time period of a complete day for all online metrics in this paper.

4.1.4 Parameter Settings. In all experiments, we use the validation set to tune the hyper-parameters to generate the best performance for different methods. The learning rate is searched from 10^{-4} to 10^{-2} . The L2 regularization term is searched from 10^{-4} to 1. All models use Adam as the optimizer. We extract micro-level interests from the user's recent 100, 50, and 20 behavior items for MovieLens, WeChat, and Taobao, respectively. For macro-level interests, we group all items into 20, 241, and 6769 clusters for MovieLens, WeChat, and Taobao, respectively. We assign each user's recent behavior to these clusters, and we select the top-5 interest groups to compute their macro-level interests.

4.2 Offline Evaluation

This section compares the experimental results of MPAD and other baselines on offline datasets to answer Q1 and Q2. First, we verify the effectiveness of interest modeling in MIE. For all datasets, we train MPAD and other competing methods using the same user behavior sequences. The interest modeling techniques differ from each other among the comparing methods. In particular, MPAD makes use of the cluster-based interest model proposed in Sec. 3.3. LR, DeepFM, and WDL treat user behavior sequences as raw features and directly feed them into linear/MLP layers for feature crossing. DIN and DIEN adopt TA/GRU units to model short-term user interests. SIM introduces an additional retrieval layer to select top- k items from user's full behavior sequences to model the lifelong user interest. As shown in Table 1, the results verify that MIE outperforms other user interest models remarkably, in terms of both AUC and Logloss. This indicates that MIE is more robust to the disturbing noise hidden in the raw item-level features within behavior sequences that may undermine learning performance.

²<https://grouplens.org/datasets/movielens/10m/>

³<https://algo.weixin.qq.com/2021/intro>

⁴<https://tianchi.aliyun.com/dataset/dataDetail?dataId=56>

Table 2: Comparison of item quality in the item list (bold; underline: runner-up). The marker * denotes that our model performs significantly better than the runner-up with $p < 0.01$ over 25 runs.

Dataset	Model	NDCG@3	NDCG@10	MAP@3	MAP@10
MovieLens	DIN	0.9017	0.9230	0.8760	0.8893
	DIEN	0.9031	0.9239	0.8775	0.8900
	SIM	0.9004	0.9219	0.8747	0.8879
	Seq2Slate	0.9098	0.9295	0.8863	0.8978
	DLCM	0.9095	0.9293	0.8857	0.8976
	PRM	<u>0.9102</u>	<u>0.9296</u>	<u>0.8865</u>	<u>0.8981</u>
	ListCVAE	0.8349	0.8800	0.8154	0.8632
	PivotCVAE	0.8608	0.8928	0.8423	0.8775
	Ours	0.9148*	0.9332*	0.8918*	0.9027*
WeChat	DIN	0.6811	0.7200	0.6084	0.5935
	DIEN	0.6970	0.7310	0.6256	0.6075
	SIM	0.6976	0.7329	0.6272	0.6100
	Seq2Slate	0.7001	0.7342	0.6309	0.6148
	DLCM	<u>0.7029</u>	<u>0.7373</u>	<u>0.6318</u>	<u>0.6164</u>
	PRM	0.7001	0.7363	0.6280	0.6152
	ListCVAE	0.5710	0.6533	0.4975	0.5333
	PivotCVAE	0.5738	0.6547	0.4993	0.5362
	Ours	0.7095*	0.7419*	0.6398*	0.6216*
Taobao	DIN	0.2017	0.3172	0.1654	0.2227
	DIEN	0.2003	0.3155	0.1631	0.2202
	SIM	0.2006	0.3200	0.1645	0.2237
	Seq2Slate	0.2093	0.3294	0.1728	0.2326
	DLCM	0.2115	<u>0.3308</u>	0.1749	<u>0.2337</u>
	PRM	<u>0.2118</u>	0.3303	<u>0.1750</u>	0.2335
	ListCVAE	0.1767	0.3076	0.1523	0.2193
	PivotCVAE	0.1785	0.3120	0.1512	0.2224
	Ours	0.2166*	0.3339*	0.1799*	0.2372*

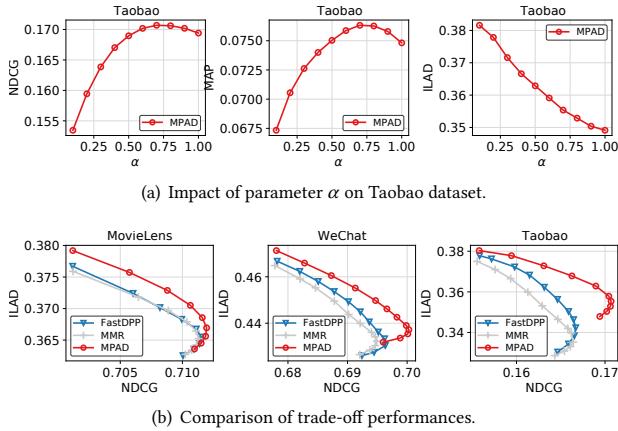


Figure 5: Trade-off between accuracy and diversity.

Next, we compare the performance of accuracy estimation among all point-wise and list-wise ranking methods. For MPAD, we only activate MIE and CAE components for this experiment. As shown in Table 2, the point-wise baselines achieve generally worse performance than the list-wise baselines on all datasets. This verifies

Table 3: Results of online A/B tests in TaoBao App.

	PV	Breadth	Stay Time	CLICK	CTR	GMV
vs Heuristic	+1.29%	+4.02%	+1.95%	+2.38%	+1.07%	+1.29%
vs fastDPP	+0.10%	+1.46%	+1.41%	+1.77%	+1.67%	+0.27%

that the mutual influence among the input ranking list incurs a great impact on list-wise recommendation. Therefore, it is of vital importance to consider the influence of browsing context in feed recommendations. Moreover, our proposed MPAD consistently yields the best performance on all datasets in terms of both NDCG and MAP. This verifies that MPAD has a superior capability to model the contextual influence among consecutive items, due to the modeling of browsing context and the user’s multi-scale interests.

Now we examine the capability to balance item accuracy and diversity in MPAD. We activate all components in MPAD for this experiment. As shown in Figure 5, when decreasing the parameter α in (6d), ILAD decreases monotonously while nDCG increases at first and then decrease a bit. When $\alpha = 0$, MPAD directly returns items with the highest accuracy scores, regardless of the item diversity. The results indicate that it is critical to introduce a proper amount of diversity into the item list to improve the joint utility of accuracy and diversity for feed recommendation. Then, we compare MPAD with MMR and fastDPP. The tunable parameters of all methods are chosen such that different algorithms have approximately the same range of nDCG. The result in Figure 5 shows that, among all comparing methods, our proposed MPAD exhibits the best item accuracy-diversity trade-off performance. This is probably due to the superior performance of accuracy and diversity estimation from MIE, CAE, and PDK. It is also noteworthy that each curve in Figure 5 has an inflection point, corresponding to the optimal balance of accuracy and diversity. In practical applications, the parameter α should be tuned to reach such an optimal status to deliver the best experience for customers.

4.3 Online Evaluation

MPAD has been fully deployed in the homepage feed of Taobao named *Guess-you-like* to serve the main traffic. In general, *Guess-you-like* one of the largest merchandise feed recommendation platform in China, which serves more than hundreds of millions of users towards billions of items every day. In *Guess-you-like*, users can slide to browse and interact with endless items in a sequential manner, as shown in Figure 1. We deploy MPAD at the re-ranking stage in *Guess-you-like* platform, which takes hundreds of candidate items from the ranking stage as input and outputs a fixed-size item list to form a new page. The online performance is compared against the fast DPP method and a heuristic method. Specifically, the fast DPP method uses point-wise ranking scores and item embedding vectors from ranking models as input, similar to [9]. The heuristic method adjusts the item order according to a series of heuristic rules predefined with expert knowledge, e.g., no more than two items within the same category on one screen. It is a commonly used diversification strategy in industrial applications.

The performance in Table 3 is averaged over two consecutive weeks. We have the following observations. Compared with the

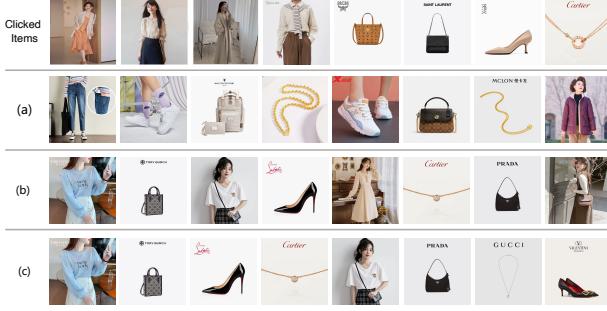


Figure 6: An example of personalized diversification.

heuristic method, first, MPAD achieves a performance improvement of 2.38% for CLICK, 0.62% for CTR, and 0.48% for GMV, indicating that our framework is able to increase the user’s willingness to interact with the items. The less improvement on GMV is due to that we mainly optimize MPAD towards the CLICK goal to be consistent with the business orientation. It is noteworthy that 1% improvement is a considerable enhancement in real-world RS, especially for applications with billion-scale users and items. In Guess-you-like, 1% improvement on CLICK brings millions of clicks every day. Second, the Category Breadth per page increases by around 4% at the same time, which verifies that MPAD is able to promote diversity in the recommended items as well as accuracy. Third, the Stay Time increases by 1.95% and the PV increases by 1.29%, which indicates that MPAD can attract users to stay at the platform. MPAD also outperforms fastDPP in all the above metrics. All these improvements verify that MPAD is able to enhance both the item accuracy and diversity in the recommendation results and well balance their trade-off to attract users in feed recommendation.

4.3.1 Case Study. In Figure. 6, we present one case to illustrate how MPAD diversifies items to suit personal interests. We sample a female customer who recently clicked a series of clothing and dressing items, which indicate her browsing interests. Figure. 6(a) presents the diversified results based on heuristic rules which are universal for all users. It is clear that a few less relevant items appear in the recommendation result, such as sports shoes and down cloth. Figure. 6(b) shows the results obtained by MPAD, where the recommended items are all relevant to the clicked items and are well-spaced to avoid presenting similar items in a row. Figure. 6(c) shows the results of adjusting the parameter α in MPAD to increase diversity. The items are now more proportioned than those in (b). For example, the number of clothing items decreases from four in (b) to only two in (c), and their distance is greater. This example qualitatively illustrates the effectiveness of MPAD in delivering perception-aware diversification services based on user interests.

5 RELATED WORK

Re-ranking Methods. Traditional point-wise ranking models focus on predicting the interaction label between any given user-item pairs, e.g., Wide&Deep [11], DIN [48] and SIM [33], regardless of the context information in a full recommendation list. However, in feed products, the mutual influence between items exhibits a great influence on user behaviors since users are reviewing items

in a sequential manner. Recent works on re-ranking propose to consider the mutual influence between items in a list-wise manner, which includes three main research lines, i.e., RNN-based methods, attention-based methods, and evaluator-generator-based methods. Specifically, the RNN-based methods model the mutual influence based on RNN structures. DLCM [2] uses gated recurrent units (GRU) to sequentially encode the top-ranked items with their feature vectors. MidNN [49] applies the long-short term memory (LSTM), with a global feature extension method to capture cross-item influences. Seq2Slate [5] extends MiDNN by adopting a more flexible pointer network to solve the re-ranking problem.

Attention-based methods use self-attention to model item interactions without RNN’s sequential structure. PRM [32] uses pre-trained embedding to extract item interactions and generate list-wise predictions with self-attention blocks and position encoding. PFRN [20] uses Listwise Feature Encoding for context-aware item interaction modeling with multi-head self-attention and relative position representation. Evaluator-generator methods use a generator to generate permutations and an evaluator to determine optimal permutation, e.g., SEG [43] and GRN [14]. These re-ranking models mainly focus on improving recommendation accuracy instead of a joint utility of both accuracy and diversity.

Diversity Methods. It has been widely acknowledged in diversified recommendation methods that accuracy should not be the only goal of recommendation tasks since it may lead to a return of highly similar items to harm user’s satisfaction with the recommendation results [1, 3, 6–9, 21, 27, 37, 41, 44, 46]. Research on diversification includes three main streams. The first stream of methods adopts heuristics rules to deal with item order in a post-processing manner. The representative work is maximal marginal relevance (MMR) [8], which represents relevance and diversity with independent metrics and maximizes the marginal relevance with a trade-off parameter. Other greedy heuristics methods vary in the definition of this marginal relevance [3, 6–8, 37, 41]. The second stream of methods treats diversified recommendation as an end-to-end learning task. DCF [12] proposes to solve the coupled parameterized matrix factorization and structural learning problems based on collaborative filtering. BGCF [42] applies bayesian graph convolutional neural networks to model the uncertainty between user-item and bring diversity into recommendation indirectly. DSSA [24] adopts the attention mechanism to determine the importance of the under-covered subtopics, where the relevance and the diversity are jointly estimated with subtopic attention. The third stream of methods is based on statistical models. The representative is the determinantal point process (DPP) which measures set diversity by describing the probability for all subsets of the item set. The maximum a posteriori (MAP) in DPP to generate diverse lists is NP-hard, such that many related works focus on the approximation of DPP for low-complex iterates. For example, Fast DPP [9] proposes a greedy approximation to accelerate the MAP inference for DPP. This fast DPP method also inspires many follow-ups to improve diversity in different recommendation tasks [15, 30]. Meanwhile, SSD [21] proposed a time series analysis technique to include out-of-window items into the measurement of diversity to increase the diversity of a long recommendation sequence and alleviate the long tail effect as well.

User Interest Modeling. Researchers are capturing shifting user interests by modeling behavior sequences. For example, DIN [48] uses TA to capture user diversity, DIEN [47] uses GRU for drifting temporal interest, and MIND [26] uses multi-vectors for dynamic interests. These models focus on short sequences (<100). For long sequences, memory-based methods such as HPMN [40] and MIMN [33] use memory networks to model diverse user interests, while two-stage methods such as SIM [34] and UBR4CTR [36] train retrieval and CTR models separately. In the first stage, the retrieval model retrieves the top- k relevant items from long user behavior sequences and stores the subsequence in an offline database. Then, in the second stage, the CTR model retrieves the top- k relevant items directly from the offline database to reduce complexity during the learning. These models mainly focus on the CTR tasks with the goal of maximizing accuracy. Their successes in CTR prediction inspire us to extract user interests from both the long and short behavior sequences.

6 CONCLUSION

In this paper, we propose a general re-ranking framework named MPAD for practical feed recommendation. A series of collaborative models are proposed to sequentially evaluate the accuracy and diversity of different items in a list and to generate an optimal item list by maximizing the joint utility of accuracy and diversity of the entire list. Both online and offline experiments verified the effectiveness of the proposed framework.

REFERENCES

- [1] Mustafa Abdool, Malay Haldar, Prashant Ramanathan, Tyler Sax, Lanbo Zhang, Aamir Manaswala, Lynn Yang, Bradley Turnbull, Qing Zhang, and Thomas Legrand. Managing diversity in airbnb search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2952–2960, 2020.
- [2] Qingyao Ai, Keping Bi, Jiafeng Guo, and W Bruce Croft. Learning a deep listwise context model for ranking refinement. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 135–144, 2018.
- [3] Azin Ashkan, Branislav Kveton, Shlomo Berkovsky, and Zheng Wen. Optimal greedy diversity for recommendation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
- [4] Michael J Barber. Modularity and community detection in bipartite networks. *Physical Review E*, 76(6):066102, 2007.
- [5] Irwan Bello, Sayali Kulkarni, Sagar Jain, Craig Boutilier, Ed Chi, Elad Eban, Xiyang Luo, Alan Mackey, and Ofer Meshi. Seq2slate: Re-ranking and slate optimization with rnns. *arXiv preprint arXiv:1810.02019*, 2018.
- [6] Rubi Boim, Tova Milo, and Slava Novgorodov. Diversification and refinement in collaborative filtering recommender. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 739–744, 2011.
- [7] Allan Borodin, Hyun Chul Lee, and Yuli Ye. Max-sum diversification, monotone submodular functions and dynamic updates. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI symposium on Principles of Database Systems*, pages 155–166, 2012.
- [8] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, 1998.
- [9] Laming Chen, Guoxin Zhang, and Eric Zhou. Fast greedy map inference for determinantal point process to improve recommendation diversity. In *Advances in Neural Information Processing Systems*, volume 31, 2018.
- [10] Qiwei Chen, Yue Xu, Changhua Pei, Shanshan Lv, Tao Zhuang, and Junfeng Ge. Efficient long sequential user data modeling for click-through rate prediction. *arXiv preprint arXiv:2209.12212*, 2022.
- [11] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10, 2016.
- [12] Peizhe Cheng, Shuaiqiang Wang, Jun Ma, Jiankai Sun, and Hui Xiong. Learning to recommend accurate and diverse items. In *Proceedings of the 26th international conference on World Wide Web*, pages 183–192, 2017.
- [13] Liang Feng, Qianchuan Zhao, and Cangqi Zhou. Improving performances of top-n recommendations with co-clustering method. *Expert Systems with Applications*, 143:113078, 2020.
- [14] Yufei Feng, Binbin Hu, Yu Gong, Fei Sun, Qingwen Liu, and Wenwu Ou. GRN: Generative rerank network for context-wise recommendation. *arXiv preprint arXiv:2104.00860*, 2021.
- [15] Lu Gan, Diana Nurbakova, Léa Laporte, and Sylvie Calabretto. Enhancing recommendation diversity using determinantal point processes on knowledge graphs. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2001–2004, 2020.
- [16] Mike Gartrell, Elvis Dohmatob, and Jon Alberdi. Deep determinantal point processes. *arXiv preprint arXiv:1811.07245*, 2018.
- [17] Hufeng Guo, Ruiming Tang, Yuning Ye, Zhenguo Li, and Xiuqiang He. Deepfm: A factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, page 1725–1731, 2017.
- [18] Qi Hao, Tianze Luo, and Guangda Huzhang. Re-ranking with constraints on diversified exposures for homepage recommender system. *arXiv preprint arXiv:2112.07621*, 2021.
- [19] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.
- [20] Jinhong Huang, Yang Li, Shan Sun, Bufeng Zhang, and Jin Huang. Personalized flight itinerary ranking at fliggy. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2541–2548, 2020.
- [21] Yanhua Huang, Weikun Wang, Lei Zhang, and Ruiwen Xu. Sliding spectrum decomposition for diversified recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3041–3049, 2021.
- [22] Kalervo Järvelin and Jaana Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *ACM SIGIR Forum*, volume 51, pages 243–250, 2017.
- [23] Ray Jiang, Sven Gowal, Yuqiu Qian, Timothy Mann, and Danilo J Rezende. Beyond greedy ranking: Slate optimization via list-cvae. In *International Conference on Learning Representations*.
- [24] Zhengbao Jiang, Ji-Rong Wen, Zhicheng Dou, Wayne Xin Zhao, Jian-Yun Nie, and Ming Yue. Learning to diversify search results via subtopic attention. In *Proceedings of the 40th international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 545–554, 2017.
- [25] Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286, 2012.
- [26] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. Multi-interest network with dynamic routing for recommendation at tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, page 2615–2623, 2019.
- [27] Zihan Lin, Hui Wang, Jingshu Mao, Wayne Xin Zhao, Cheng Wang, Peng Jiang, and Ji-Rong Wen. Feature-aware diversified re-ranking with disentangled representations for relevant recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3327–3335, 2022.
- [28] Shuchang Liu, Fei Sun, Yingqiang Ge, Changhua Pei, and Yongfeng Zhang. Variation control and evaluation for generative slate recommendations. In *Proceedings of the Web Conference 2021*, pages 436–448, 2021.
- [29] Weiwei Liu, Yunjia Xi, Jiarui Qin, Fei Sun, Bo Chen, Weinan Zhang, Rui Zhang, and Ruiming Tang. Neural re-ranking in multi-stage recommender systems: A review. *arXiv preprint arXiv:2202.06602*, 2022.
- [30] Yong Liu, Yingtao Xiao, Qiong Wu, Chunyan Miao, Juyong Zhang, Binqiang Zhao, and Haihang Tang. Diversified interactive recommendation with implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4932–4939, 2020.
- [31] H Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230, 2013.
- [32] Changhua Pei, Yi Zhang, Yongfeng Zhang, Fei Sun, Xiao Lin, Hanxiao Sun, Jian Wu, Peng Jiang, Junfeng Ge, Wenwu Ou, et al. Personalized re-ranking for recommendation. In *Proceedings of the 13th ACM conference on recommender systems*, pages 3–11, 2019.
- [33] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2671–2679, 2019.

- [34] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, page 2685–2692, 2020.
- [35] Xufeng Qian, Yue Xu, Fuyu Lv, Shengyu Zhang, Ziwen Jiang, Qingwen Liu, Xiaoyi Zeng, Tat-Seng Chua, and Fei Wu. Intelligent request strategy design in recommender system. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3772–3782, 2022.
- [36] Jiarui Qin, Weinan Zhang, Xin Wu, Jiarui Jin, Yuchen Fang, and Yong Yu. User behavior retrieval for click-through rate prediction. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 2347–2356, 2020.
- [37] Lijing Qin and Xiaoyan Zhu. Promoting diversity in recommendation by entropy regularizer. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, page 2698–2704, 2013.
- [38] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining*, pages 1149–1154, 2016.
- [39] C. E. Rasmussen and C. I. K. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [40] Kan Ren, Jiarui Qin, Yuchen Fang, Weinan Zhang, Lei Zheng, Weijie Bian, Guorui Zhou, Jian Xu, Yong Yu, Xiaoqiang Zhu, and Kun Gai. Lifelong sequential modeling with personalized memorization for user response prediction. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 565–574, 2019.
- [41] Chaofeng Sha, Xiaowei Wu, and Junyu Niu. A framework for recommending relevant and diverse items. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, page 3868–3874, 2016.
- [42] Jianing Sun, Wei Guo, Dengcheng Zhang, Yingxue Zhang, Florence Regol, Yaochen Hu, Huirong Guo, Ruiming Tang, Han Yuan, Xiuqiang He, et al. A framework for recommending accurate and diverse items using bayesian graph convolutional neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2030–2039, 2020.
- [43] Fan Wang, Xiaomin Fang, Lihang Liu, Yaxue Chen, Jiucheng Tao, Zhiming Peng, Cihang Jin, and Hao Tian. Sequential evaluation and generation framework for combinatorial recommender system. *arXiv preprint arXiv:1902.00245*, 2019.
- [44] Mark Wilhelm, Ajith Ramanathan, Alexander Bonomo, Sagar Jain, Ed H Chi, and Jennifer Gillenwater. Practical diversified recommendations on youtube with determinantal point processes. In *International Conference on Information and Knowledge Management*, pages 2165–2173, 2018.
- [45] Mi Zhang and Neil Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 123–130, 2008.
- [46] Yu Zheng, Chen Gao, Liang Chen, Depeng Jin, and Yong Li. DGCN: Diversified recommendation with graph convolutional networks. In *Proceedings of the Web Conference 2021*, pages 401–412, 2021.
- [47] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. In *Deep Interest Evolution Network for Click-through Rate Prediction*, 2019.
- [48] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1059–1068, 2018.
- [49] Tao Zhuang, Wenwu Ou, and Zhirong Wang. Globally optimized mutual influence aware ranking in e-commerce search. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, page 3725–3731, 2018.

where the row vector c_i and the scalar $d_i \geq 0$ satisfies

$$Vc_i^\top = D_{S,i}^u, \quad (23a)$$

$$d_i^2 = D_{ii}^u - \|c_i\|_2^2. \quad (23b)$$

According to (22), we have

$$\det(D_{S \cup \{i\}}^u) = \det(VV^\top) \cdot d_i^2 = \det(D_S^u) \cdot d_i^2. \quad (24)$$

Combine (6c) with (24), we obtain

$$j = \arg \max_{i \in I \setminus S} g(u, i | S) + \alpha \cdot \log(d_i^2). \quad (25)$$

We follow [9] to derive the update of $\log(d_i^2)$ as follows. The Cholesky decomposition of $D_{S \cup \{j\}}^u$ can be written as

$$D_{S \cup \{j\}}^u = \begin{bmatrix} V & 0 \\ c_j & d_j \end{bmatrix} \begin{bmatrix} V & 0 \\ c_j & d_j \end{bmatrix}^\top. \quad (26)$$

Define c'_i and d'_i as the new vector and scalar of $i \in I \setminus (S \cup \{j\})$ after adding item j into S . According to (23a) and (26), we have

$$\begin{bmatrix} V & 0 \\ c_j & d_j \end{bmatrix} c'^{\top}_i = D_{S \cup \{j\},i}^u = \begin{bmatrix} D_{S,i} \\ D_{ji} \end{bmatrix}. \quad (27)$$

Combining (27) with Eq. (23a), we have

$$c'_i = \begin{bmatrix} c_i & (D_{ji}^u - \langle c_j, c_i \rangle) / d_j \end{bmatrix} \doteq \begin{bmatrix} c_i & e_i \end{bmatrix}. \quad (28)$$

Then (23b) implies

$$d'^2 = D_{ii}^u - \|c'_i\|_2^2 = D_{ii}^u - \|c_i\|_2^2 - e_i^2 = d_i^2 - e_i^2. \quad (29)$$

A DERIVATION OF ITEM SELECTION

The composite kernel D_S^u is a PSD matrix since it is an addition of multiple PSD elementary kernels. The Cholesky decomposition of D_S^u can be written as $D_S^u = VV^\top$, where $V \in \mathbb{R}^{k \times k}$ is an invertible lower triangular matrix. For any $i \in I \setminus S$, the Cholesky decomposition of $D_{S \cup \{i\}}^u$ can be represented as

$$D_{S \cup \{i\}}^u = \begin{bmatrix} D_S^u & D_{S,i}^u \\ D_{i,S}^u & D_{ii}^u \end{bmatrix} = \begin{bmatrix} V & 0 \\ c_i & d_i \end{bmatrix} \begin{bmatrix} V & 0 \\ c_i & d_i \end{bmatrix}^\top, \quad (22)$$