Contextual MAB Oriented Embedding Denoising for Sequential Recommendation

Zhichao Feng
School of Computer Science, Beijing University of Posts and Telecommunications
Beijing, China
fengzc@bupt.edu.cn

Pengfei Wang*
School of Computer Science, Beijing University of Posts and Telecommunications
Beijing, China
wangpengfei@bupt.edu.cn

Kaiyuan Li
School of Computer Science, Beijing University of Posts and Telecommunications
Beijing, China
tsotfsk@bupt.edu.cn

Chenliang Li
School of Cyber Science and Engineering, Wuhan University
Wuhan, China
cllee@whu.edu.cn

Shangguang Wang
School of Computer Science, Beijing University of Posts and Telecommunications
Beijing, China
sgwang@bupt.edu.cn

ABSTRACT

Deep neural networks now have become the de-facto standard for sequential recommendation. In the existing techniques, an embedding vector is assigned for each item, encoding all the characteristics of the latter in latent space. Then, the recommendation is transferred to devising a similarity metric to recommend user's next behavior. Here, we consider each dimension of an embedding vector as a (latent) feature. Though effective, it is unknown which feature carries what semantics toward the item. Actually, in reality, this merit is highly preferable since a specific group of features could induce a particular relation among the items while the others are in vain. Unfortunately, the previous treatment overlooks the feature semantic learning at such a fine-grained level. When each item contains multiple latent aspects, which however is prevalent in real-world, the relations between items are very complex. The existing solutions are easy to fail on better recommendation performance. It is necessary to disentangle the item embeddings and extract credible features in a context-aware manner.

To address this issue, in this work, we present a novel **Co**ntextual **M**AB based **E**mbedding **D**enoising model (ComeD for short) to adaptively identify relevant dimension-level features for a better recommendation. Specifically, ComeD formulates the embedding denoising task as a Contextual Multi-armed Bandit problem. For each feature of the item embedding, we assign a two-armed neural bandit to determine whether the constituent semantics should be preserved, rendering the whole process as embedding denoising. By aggregating the denoised embeddings as contextual information, a reward function deduced by the similarity between the historical interaction sequence and the target item is further designed to approximate the maximum expected payoffs of bandits for efficient learning. Considering the possible inefficiency of training the serial operating mechanism, we also design a swift learning strategy to accelerate the co-guidance between the renovated sequential embedding and the parallel actions of neural bandits for a better recommendation. Comprehensive trials conducted on four widely recognized benchmarks substantiate the efficiency and efficacy of our framework.

CCS CONCEPTS

• **Information systems → Recommender systems**;

KEYWORDS

Sequential Recommendation; Contextual Multi-Armed Bandit; Representation Denoising

ACM Reference Format:
Zhichao Feng, Pengfei Wang, Kaiyuan Li, Chenliang Li, and Shangguang Wang. 2024. Contextual MAB Oriented Embedding Denoising for Sequential Recommendation . In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM '24), March 4–8, 2024, Merida, Mexico.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3616855.3635798

1 INTRODUCTION

Deep neural networks have emerged as the preeminent and triumphant methodologies for addressing the sequential recommendation task, as evidenced by their widespread adoption and remarkable proficiency in representation learning [3, 6, 9, 16]. Conventionally, these methodologies entail the acquisition of an item's representation in the form of a densely packed vector, and an intricate neural architecture is crafted to extrapolate users' intentions. After that, a discerning similarity metric is further employed to systematically rank the assortment of potential candidate items for recommendation.

While undoubtedly efficacious, such an approach compels the assimilation of characteristics to conform uniformly to a spectrum of item relationships. This coarse alignment could potentially lead to sub-optimal outcomes when attempting to fit intricate item relationships [21]. In practical settings, item attributes are distinctly designated (i.e., fabric type, care instructions, and closure type of a

*Corresponding author.

WSDM '24, March 4–8, 2024, Merida, Mexico

ACM ISBN 979-8-4007-0371-3/24/03…$15.00
https://doi.org/10.1145/3616855.3635798

shirt, etc), thus a specific group of features could induce a particular relation among the items, whereas the remaining attributes may exert a comparatively marginal influence. Ideally, a better representation learning metric should possess the capacity to elucidate its semantics, enabling it to adeptly adapt to diverse item relationships [2, 40], especially for the sequential recommendation task, usually, a series of interacted items is concerned, the accumulation of irrelevant characteristics will inevitably lead to low performance and interpretability. To explain this, we give an illustrative example, as Fig. 1 shows.

It is evident that the item candy exhibits dual, distinct semantic clusters. The dietary semantics, exemplified by its relevance to occasions like tea parties, find application in the sub-sequence chocolate → candy → cake. In contrast, the festive semantics, pertaining to events such as Halloween celebrations, seamlessly align with the pattern pumpkin → mask → candy → wizard hat. In comparison to the unified approach that considers all characteristics to infer users' intentions, the disentangled approach over these features proves more suitable for accommodating different relations without introducing any irrelevant features.

While the concept may be alluring in theory, the implementation of an efficacious denoising strategy from the feature level, devoid of any supervision signals, poses significant challenges. Furthermore, the proliferation of data dimensions associated with an interacted item sequence adds complexity to the selection process, rendering the exploration of the vast dimension space infeasible for the feature-level denoising task. Recently, some attention-based approaches are proposed [22, 46] to automatically assign different weights to each feature in terms of their relevance estimation. Although these methods aim to address the issue of relevance, the accumulation of potential noise can still considerably complicate the sequence learning process. Automatically selecting the appropriate characteristics of each item representation and effectively integrating them to enhance recommendation systems represents an intriguing yet formidable task.

To address this issue, in this work, we present a novel **Co**ntextual **M**AB based **E**mbedding **D**enoising Model (ComeD for short) of denoising item representations from feature level for a better recommendation. Specifically, ComeD formalizes the representation denoising task as a Contextual Multi-armed Bandit problem. For each characteristic of item representations, we assign a two-armed neural bandit, and each arm is associated with a fixed but unknown reward probability distribution to determine whether the corresponding characteristic should be preserved. Based on the selected arms, we adeptly aggregate items based on their preserved characteristics to forge a superior-quality sequential representation. A reward function map deduced by the similarity between sequential embedding and the target item is further designed to approximate the maximum expected payoff of each bandit to empower an efficient recommendation process. To mitigate the computational burden arising from the serial learning strategy on expansive dimensions, we further introduce a swift learning strategy to derive a co-guidance between the renovation of sequential information and the parallel collaboration of neural bandits for efficient learning: In the initial phase, we simultaneously feed the sequential embedding into each bandit, guiding their arm selection in parallel. Subsequently, we augment the obtained sequential embedding by

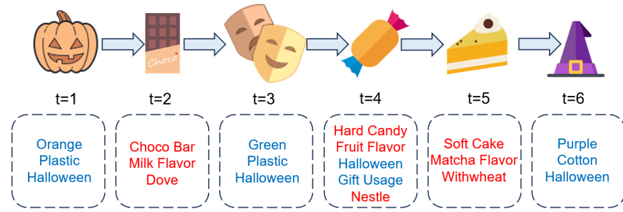

Figure 1: An example to explain the different functions that characteristics contributed for recommendation. The red dimension represents characteristics relevant to the pattern *chocolate* → *candy* → *cake*, while the blue one represents the characteristics compatible with the pattern *pumpkin* → *mask* → *candy* → *wizard hat*.

discarding irrelevant characteristics from the item representations based on the chosen arms. We cyclically iterate through these two phases to progressively approach the attainment of an optimally refined sequential embedding for efficient learning.

We conduct comprehensive experiments on four datasets with a range of competitive baselines for a rigorous evaluation. Experimental results show that our proposed ComeD can significantly outperform all the baselines in multiple metrics. To summarize, the contributions of this paper are listed as follows:

- We formalize the characteristic denoising task into a Contextual Multi-armed Bandit problem. To the best of our knowledge, this is the first work of performing the denoising from the feature level in a hard-coding manner specifically tailored for sequential recommendation.
- We design a reward assignment mechanism deduced by the similarity between the renovated sequential embedding and the target item to approximate the maximum payoff of each action for recommendation.
- We introduce a swift learning strategy to derive the co-guidance between the parallel corporations of all bandits and the renovation of sequential embeddings, resulting in a more efficient learning process.
- Extensive experiments on four real-world datasets demonstrate that our model obtains significant performance improvement over both traditional and denoising-enhanced sequential recommenders.

2 RELATED WORK

In this section we present a concise overview of the pertinent literature through three distinct viewpoints: sequential recommendation, contextual bandit recommendation, and feature selection techniques.

2.1 Sequential Recommendation

Sequential recommendation focuses on how to effectively model the sequence of interacted items. With the development of deep models, RNN-based [10, 16] and CNN-based [33, 34, 50] networks have been widely utilized to model the interacted sequences. More recently, attention-based [14, 17, 32] and GNN-based models [11, 44, 48] are applied to the sequential recommendation and proved to be advantageous in discovering user behavior patterns due to the ability to capture more complex item transitions. However, most methods mainly focus on the item-level interaction history. Recently, some pioneering works adopt attention-based networks [22, 51] to model item relations from the feature level, though

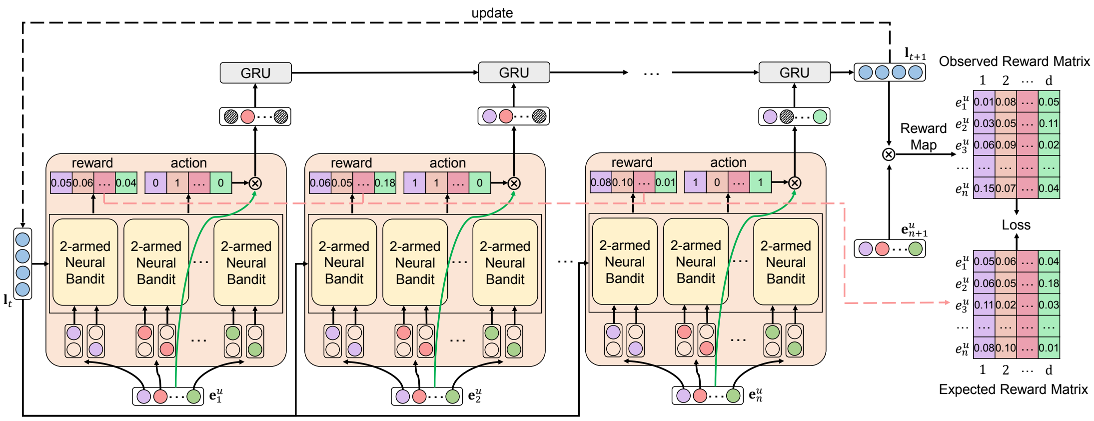

Figure 2: The overall architecture of Contextual MAB based Embedding Denoising (ComED for short) of learning denoising representations.

effective, these approaches still fail to obtain the truly relevant features for further consideration, which limits the recommendation performance and interpretability.

2.2 Contextual Bandit Recommendation

The contextual bandit model refers to the selection of the best arm with the presence of context features in a multi-armed bandit. In recommender system, existing works mainly focus on how to utilize the user/item context available for the contextual bandit to provide a more effective recommendation. Early efforts are devoted to using explicit user and item information as a context vector [7, 19, 30, 31, 56]. For example, Zhu et al. [56] took all the item features as contexts, while Gutowski et al. [7] took user profiles and historical behaviors as contexts for a personalized recommendation. More recently, there are also some works focusing on extracting implicit features as contexts [5, 8, 39, 43]. For example, Wu et al. [43] utilized statistical methods to convert tag information into embeddings as context vectors. Wang et al. [39] obtained the implicit context features through the matrix factorization of the user-item rating matrix. While Shen et al. [29] simultaneously integrated implicit features and explicit features for streaming recommendation. In our work, latent representations of sequences and characteristics are used for context vectors and guide the selection of characteristics.

2.3 Feature Selection Techniques

Feature selection refers to a class of computational methods where the goal is to select a subset of useful features from the original feature set in a dataset. These methods can largely be divided into filters, wrappers, and embedded methods [45]. The filter methods usually selected features based on information theoretical measures that do not optimize the model itself [28]. For wrapper methods, the results of models will be used as the evaluation criteria for the feature selection. Due to the time complexity of the wrapper method, few works pay attention to this method. While in embedded methods, the feature selection and model learning are optimized simultaneously which will consider both efficiency and performance [12, 13, 35, 41, 42].

Recently, disentangled representations of items have received increasing attention in recommendation systems, which can also be regarded as an embedded method for feature selection. These works mainly focus on assigning weights on features [20] or disentangling features from aspect level [2, 4]. In contrast to their study, our focus is directed to perform fine-grained feature-level denoising for item representations automatically in a hard coding mechanism for a better recommendation.

3 PRELIMINARY

Notations. Let $\mathcal{U}$ denote a set of users and $\mathcal{E}$ denote a set of items, where $|\mathcal{U}|$ and $|\mathcal{E}|$ are the numbers of users and items. For each user $u \in \mathcal{U}$, we use $e^u_{1:n} = \{e^u_1, e^u_2, \cdots, e^u_i, \cdots, e^u_n\}$ to represent the interaction sequence of items, where $e^u_i \in \mathcal{E}$. The task of sequential recommendation is to infer the likelihood of each target item e^u_{n+1} that the user will purchase at the next visit.

Contextual Multi-armed Bandit. The Contextual Multi-armed Bandit is a classic reinforcement learning problem that has non-associative states and focuses only on valuable feedback. Formally, for each round $t \in T$, the agent observes the context consisting of several feature vectors. Based on the observed contexts, it performs action a_t (i.e. chooses an arm) , and then receives the corresponding reward r_{a_t}. The objective is to maximize its expected total reward over T rounds, which is equivalent to minimizing the following pseudo-regret:

$$R_T = \mathbb{E}\left[\sum_{t=1}^{T}(r_{a^*_t} - r_{a_t})\right] \tag{1}$$

where a^*_t is the action with a maximum expected payoff at t round.

4 METHODOLOGY

In this section, we first introduce the proposed **Co**ntextual **M**AB based **E**mbedding **D**enoising model (ComED for short) in detail, the architecture is shown in Fig. 2. In particular, we analyze how to formalize feature denoising task for sequential recommendation into the Contextual Multi-armed Bandit problem, then give detail of its learning and prediction procedure. We subsequently conducted an in-depth analysis to elucidate the indispensability and merits associated with the utilization of a multi-armed bandit. In the following, we will drop superscript u when there is no ambiguity.

4.1 Contextual Bandit for Featuring Denoising

Given a series of interaction data $e_{1:n}$, we utilize a contextual MAB to automatically extract the informative features for recommendation. Specifically, let $\mathbf{e}_i \in R^d$ denote the raw embedding vector of the interacted item e_i in a d-dimensional continuous space. By assigning a two-armed neural bandit for each dimension, ComED scans all dimensions of the interacted sequence serially to determine whether the corresponding features should be preserved. The arm space is $a \in \{pos = 1, neg = 0\}$, where we use arm pos=1 to indicate that the corresponding feature should be preserved, and arm neg=0 indicates that the feature is irrelevant to the current context. Inspired by [49], the neural bandit responsible for the k-th dimension of e_i at t-th round performs actions (i.e. chooses an arm) according to the upper confidence bound:

$$a_t = \arg\max_{a \in \{0,1\}} \{\mathbf{w} \cdot MLP(\mathbf{x}_{a_t}) + \gamma ||MLP(\mathbf{x}_{a_t})||_{A_t^{-1}})\} \tag{2}$$

where $\mathbf{x}_{a_t}$ represents the feature vector of context-arm pair for arm a at round t, $\mathbf{w}$ is a vector that maps the transformed vector into the expected reward, $A_t = \lambda \mathbf{I} + \sum_t \mathbf{x}_{a_t} \mathbf{x}_{a_t}^T$ is a matrix based on the historical chosen context-arm pairs, and γ is a tuning parameter that controls the exploration rate. In our work, we follow [53] to encode the contextual information $\mathbf{x}_{a_t}$ as follows:

$$\mathbf{x}_{a_t} = \begin{cases} \mathbf{l}_{t-1} \oplus [\mathbf{e}_{i,k}, 0] \text{when } a = 1 \\ \mathbf{l}_{t-1} \oplus [0, \mathbf{e}_{i,k}] \text{when } a = 0 \end{cases} \tag{3}$$

where $\oplus$ is a concatenation operator, $\mathbf{l}_{t-1}$ represents the sequential embedding of $e_{1:n}$ keeping the preserved characteristics of items at (t-1)-th round, and $\mathbf{e}_{i,k}$ represents the k-th element of $\mathbf{e}_i$. Note that when presented with the input $e_{1:n}$, ComeD needs to ascertain the status of characteristics associated with each item successively. To clarify the connection between action step and the item dimensions, we give a rather straightforward formula, where $i = (t/d)\%n + 1$, and $k = t\%d$ at t-th round.

Once the action at t-th round is performed according to Eq. 2, ComeD updates the item representation immediately (e.g., preserves or sets the corresponding feature to 0). After that, a renewed sequential embedding $\mathbf{l}_t$ is generated for the next arm selection based on the updated item representations. To simplify, in this work, we utilize a GRU unit [10] to aggregate the item representations. It's worth noting that for the initial time step (t = 1), $\mathbf{l}_0$ retains the complete spectrum of item attributes.

4.2 Reward Map Assignment Strategy

Recall that according Eq. 2, ComeD performs an action, the triggered reward $r_{a_t} = \mathbf{w} \cdot MLP(\mathbf{x}_{a_t})$ is obtained. However, the observed reward $\hat{r}_{a_t}$ is infeasible due to the lack of supervision signals, making ComeD fails to judge the specific quality of the arm selection for each bandit. Considering previous works [18] usually treat the similarity metric between sequential information and the target item as an informative global reward to guide the learning procedure. To solve the missing observed reward each bandit encountered, in the following, we aim to derive each observed reward reversely according to the obtained similarity score.

Specifically, when the action a_t on the k-th dimension of e_i is performed, the value of $\mathbf{e}_{i,k}$ is tuned based on the selected arm: if a_t=1, then $\mathbf{e}_{i,k}$ remains unchanged, else we discard it by setting $\mathbf{e}_{i,k}$=0. After that, we recalculate the sequential embedding based on the changed item representation $\mathbf{e}_i$ according to the GRU unit and obtain $\mathbf{l}_t$. Based on the renovated sequential embedding $\mathbf{l}_t$ caused by action a_t, we calculate the similarity between $\mathbf{l}_t$ and the target item $\mathbf{e}_{n+1}$ and the function is written as follows:

$$s(\mathbf{l}_t, \mathbf{e}_{n+1}) = \frac{\mathbf{l}_t \cdot \mathbf{e}_{n+1}}{||\mathbf{l}_t|| \cdot ||\mathbf{e}_{n+1}||} \tag{4}$$

After that, based on the $s(\mathbf{l}_t, \mathbf{e}_{n+1})$ obtained, we use the following function to deduce an observed reward $\hat{r}_{a_t}$ that action a_t corresponds:

$$\begin{cases} \hat{r}_{a_t} = \alpha_i \cdot \beta_k \cdot s(\mathbf{l}_t, \mathbf{e}_{n+1}) \\ \alpha_i = \dfrac{exp(\mathbf{e}_i \cdot \mathbf{e}_{n+1})}{\sum_i exp(\mathbf{e}_i \cdot \mathbf{e}_{n+1})}; \beta_k = \dfrac{exp(\mathbf{e}_{i,k} \cdot \mathbf{e}_{n+1,k})}{\sum_k exp(\mathbf{e}_{i,k} \cdot \mathbf{e}_{n+1,k})} \end{cases} \tag{5}$$

According to such a design, $s(\mathbf{l}_t, \mathbf{e}_{n+1})$ is partitioned into a $n \times d$ size matrix, where the rows correspond to the items in the interacted sequence $e_{1:n}$, and columns represent d dimensions of item representations. Each cell of this formulated matrix denotes an estimated observed reward linked to a specific bandit, functioning akin to an attention score that subtly modulates the selection of individual dimensions to facilitate an accurate recommendation.

Note that when $t\%(n \times d) = 0$, it demonstrates ComeD has finished the denoising process for the whole interacted sequence $e_{1:n}$, and item representations are reset to its raw representations ready for the next denoising process.

4.3 Learning and Discussion

4.3.1 Learning. Based on the action and reward defined in the previous section, we then minimize the following empirical loss function:

$$\mathcal{L} = \sum_t^T \left(\mathbf{w} \cdot MLP(\mathbf{x}_{a_t}) - \hat{r}_{a_t}\right)^2 \tag{6}$$

Given the loss function, we follow [53] to train our model through the utilization of gradient descent. Nonetheless, the acquisition of training instances presents challenges stemming from the subsequent two factors: ComeD assigns a neural bandit for each feature in the interacted sequence, the huge count causes ComeD to need various bandits to manage arms; besides, ComeD needs to renovate sequential embedding and rewards for each performed action; and a complete denoising procedure for an interacted sequence requires scanning all dimensions of interacted items for recommendation. All of these make ComeD difficult to train efficiently.

To address this issue, we further design a swift learning strategy to learn ComeD efficiently. Specifically, We facilitate the convergence of identical item dimensions through the utilization of a shared neural bandit for parameter allocation. In accordance with this design, the ComeD model adeptly oversees a total of d distinct multi-armed bandits. After this, different from the previous metric of maintaining one bandit for each round, inspired by [25] of utilizing bandits for batch-wise action execution simultaneously, we directly feed the sequential embedding to each bandit as a guide and direct their actions in parallel according to Eq. 2 for acceleration. Based on actions performed by all bandits, we refresh embeddings of all interacted items and in turn, renovate the sequential embedding. We iterate through these two co-guidance phases for m steps to approximate the optimal sequential embedding.

In the recommendation procedure, with the learned ComeD, given a user and his/her interaction sequence, we follow the training procedure to obtain the refined sequential embedding, and feed it into Eq. 4 to rank the candidate items. We select the top-N results as the final recommendation.

4.3.2 Discussion. In this part, we give a analysis to describe the advantage of utilizing MAB for a fine-grained feature-level denoising task. First, due to the inherent absence of a supervision signal in

Table 1: Statistics of the datasets.

Dataset	#Users	#Items	#Interactions	Sparsity%
Beauty	22,363	12,101	198,502	99.93
Sports	25,598	18,357	296,337	99.95
Movies	14,260	11,868	627,164	99.63
Tmall	26,308	18,634	574,936	99.88

this particular task, attaining optimal performance becomes inherently challenging. While MAB algorithm[1] has the advantage of being able to learn from limited feedback. This characteristic makes it particularly well-suited for the current scene, where efficient exploration of different arm options is crucial for achieving optimal optimization. Furthermore, the MAB algorithm offers a solution to the challenges posed by hard coding and non-differentiable gradients. By employing an end-to-end gradient descent approach, the MAB algorithm enables seamless model updates. Another interesting benefit is that in COMED we assign a two-armed bandit for each dimension of item representations. If we enlarge the dimension scope of the bandit maintained (e.g., each bandit manages several dimensions), then COMED transforms to an aspect-aware denoising model [2], and if we assign one bandit to manage all dimensions of an item, COMED becomes an item-level denoising model, which is quite similar to [36]. This affords COMED a remarkable degree of adaptability, through the manipulation of the dimensions under a bandit's jurisdiction, which is able to establish connections with diverse denoising techniques. This flexibility renders COMED a highly versatile and accommodating framework.

5 EXPERIMENTS

In this section, we evaluate our model by comparing it with many state-of-the-art sequential recommenders, including the traditional sequential models, the denoising-enhanced sequential models, and the attribute-augmented ones. We begin by introducing the experimental setup and analyze the experimental results.

5.1 Experimental Setup

Dataset. We use four real-world datasets in our experiments, i.e. Amazon[1] and Tmall[2]. For Amazon, we select its three widely used categories, which are Beauty, Sports and Outdoors (short for Sports), and Movies and TV (short for Movies). The statistics of the four datasets are shown in Table 1. For these datasets, we first group interactions by users. After that, we filter users and items with fewer than five interaction records for Beauty and Sports, while set a threshold of fifteen [24] for users and ten for items for Movies and Tmall datasets due to their size. For each user, we sort her/his records according to the timestamp to form a user behavior sequence. we hold out the last item of each user as test data, the penultimate one as validation data, and the rest as training data.

Compared Models. To evaluate the effectiveness of our approach, We compare our model against three types of baselines, including three traditional sequential models, five denoising-enhanced sequential models, and two attribute-augmented sequential models. Specifically, we consider three traditional sequential-based models:

- **GRU4Rec**[10] is a widely used session-based recommendation based on a Gated Recurrent Unit.
- **SASRec**[14] is a self-attentive model. It utilizes a unidirectional multi-head attention mechanism to model the whole sequence.
- **BERT4Rec**[32] utilizes a bidirectional multi-head attention mechanism to model item relations.

For denoising-enhanced sequential models, we consider the following five baselines:

- **RAP** [36] is a pattern-enhanced contrastive policy learning network of performing denoising from item level.
- **FMLP-Rec** [55] proposes a learnable filter to weaken the noise in the frequency domain.
- **HGN**[22] performs denoising on the interacted sequence from both instance-level and feature-level through a hierarchical gating network for recommendation.
- **ARD**[2] is a aspect re-distribution model. It calculates the weights of different aspects to reconstruct the item embeddings.
- **ARD$^+$** is an extension of ARD by directly deleting the aspects with their weights smaller than a given threshold designed by us. Compared with ARD, ARD$^+$ applies a hard-coding metric of ignoring aspects with smaller weights.

For attribute-augmented sequential models, we consider CARCA and a modified version of COMED:

- **CARCA**[27] is a attribute-augmented recommendation model that utilizes the attention mechanism to dynamically assign weights on item attributes.
- **COMED$^+$** is an extension of COMED by feeding extra item attributes into COMED for a deniosed recommendation.

Evaluation Metric. We employ top-N Recall (Recall@N), and top-N Normalized Discounted Cumulative Gain (NDCG@N), which are widely adopted within the realm of relevant studies (with N being set to 5 or 10). Following the common strategy [15, 38, 47], we pair the ground-truth item with 1000 randomly sampled negative items that the user has not interacted with considering both the computation efficiency and the estimation quality. We calculate all metrics according to the ranking of the items and report the final result.

Parameter Settings. For a fair comparison, we adopt the following setting for all methods: the batch size is set to 256; all embedding parameters are randomly initialized in the range of(0,1). For the other parameters, we follow the settings in the original paper or source code. For FMLP-Rec[3] and CARCA[4], we use the source code provided by their authors. For other methods, we implement them by RecBole [52]. We optimize them according to the validation sets. For COMED, we implement it based on PyTorch with Adam optimizer. The exploration factor in Eq. 2 is set to 0.02, the iteration step m used in the swift learning strategy is configured at 3, and the embedding size is set to 64.

In the context of attribute-augmented models, we leverage both item images and attributes for the Amazon dataset, whereas for the Tmall dataset, our focus is solely on item attributes owing to the absence of images. Specifically, in the case of the Amazon dataset, we apply PCA to standardize the image embedding size [18], and take into account item attributes including category, brand, and

[1] 1http://jmcauley.ucsd.edu/data/amazon/
[2] https://tianchi.aliyun.com/dataset/53
[3] https://github.com/RUCAIBox/FMLP-Rec
[4] https://github.com/ahmedrashed-ml/CARCA

Table 2: Performance comparison between our ComeD and baselines (all values in the table are percentage numbers with % omitted). The best performance of each field is highlighted in boldface. Results denoted by ∗ represent the best baseline. Symbol ▲ denotes the relative improvement of ComeD against the best baseline that are consistently significant at 0.05 level.

Dataset	Metrics	Sequential Models			Denoising-enhanced Sequential Models						▲	Attribute-augmented Seuquential Models		▲
		GRU4Rec	SASRec	BERT4Rec	RAP	FMLP-Rec	HGN	ARD	ARD$^+$	ComeD		CARCA	ComeD$^+$	
Beauty	Recall@5	10.87	15.27	15.80	16.01	15.92	13.42	16.30	16.60	18.10	9.04	18.50*	**19.25**	4.05
	NDCG@5	7.05	10.96	11.34	11.56	11.28	9.46	12.13	12.35	13.48	9.15	13.23*	**14.26**	7.79
	Recall@10	16.64	20.98	21.32	22.28	22.15	18.27	22.12	22.38	23.52	4.78	24.32*	**25.12**	3.29
	NDCG@10	8.91	12.80	13.12	13.53	13.34	11.02	13.94	14.15	15.24	7.70	15.02*	**15.92**	5.99
Sports	Recall@5	8.84	11.09	11.32	11.69	12.27	10.45	12.98	13.21	14.24	7.80	14.72*	**15.36**	4.34
	NDCG@5	5.87	7.72	7.97	8.58	8.36	7.16	8.98	9.03	9.56	5.87	10.02*	**10.46**	4.39
	Recall@10	13.10	15.83	16.10	16.83	17.82	14.89	17.60	17.35	19.43	9.14	19.16*	**20.23**	5.58
	NDCG@10	8.11	9.98	10.10	10.59	10.15	9.11	10.98	11.12	11.87	6.45	12.05*	**12.78**	6.06
Movies	Recall@5	16.82	18.55	18.77	19.34	19.89	18.49	20.11	20.32	21.55	5.52	22.37*	**23.18**	3.62
	NDCG@5	11.11	12.64	13.12	13.46	13.58	12.66	14.02	14.21	15.03	5.77	15.58*	**16.11**	3.40
	Recall@10	24.89	26.78	27.02	27.11	27.69	25.59	27.54	27.81	28.83	3.67	28.43*	**29.56**	3.97
	NDCG@10	13.72	15.20	15.36	16.03	16.37	14.95	16.51	16.83	17.51	4.04	18.06*	**18.72**	3.65
Tmall	Recall@5	19.44	22.64	21.54	22.89	23.43	19.30	23.58	23.71	25.14	6.62	24.78*	**25.97**	4.80
	NDCG@5	14.02	16.39	15.81	16.58	17.13	14.79	18.02	18.15	20.11	10.80	19.66*	**20.86**	6.10
	Recall@10	25.18	28.81	28.69	28.91	29.82	23.69	29.12	29.31	29.95	0.40	29.64*	**30.54**	3.03
	NDCG@10	15.88	18.39	18.10	19.21	19.58	16.21	19.92	20.09	21.47	6.87	21.78*	**22.63**	3.90

price. Additionally, for the Tmall dataset, we incorporate attributes encompassing categories and seller stores into our considerations. These extra features are then integrated with the item embeddings to enable the denoised recommendation process.

5.2 Performance Comparison

We compare the performance of our model with the baselines. The overall performance of our proposed ComeD and the baselines are reported in Table 2. We have the following observations:

For traditional sequential recommenders, we can observe that GRU4Rec performs worst. By utilizing an attention mechanism to highlight the useful item relation softly, we find that the attention-based sequential models (SASRec and BERT4Rec) perform better than the non-attention-based ones. This observation demonstrates the necessity of discerning the relevance between items for a better recommendation. Similar results can be found in [23, 37, 54].

As a denoising-enhanced sequential recommender, HGN performs better than GRU4Rec, demonstrating the feasibility of performing denoising from both item and feature levels. However, an interesting observation is that the traditional attention-enhanced recommender SASRec and Bert4Rec perform better than the denoising-enhanced HGN. This phenomenon could potentially be attributed to HGN's utilization of an average pooling mechanism for consolidating the filtered information. Such an approach might inadvertently mitigate the distinct influences of the retained attributes, consequently resulting in suboptimal performance outcomes. Besides, RAP achieves comparable performance with FMLP-Rec in all datasets, demonstrating the effectiveness of the item-level denoising strategy for sequential recommendation. However, their performance is inferior to ARD, which weights the significance of aspects more granular through a soft attention mechanism. By removing irrelevant aspects according to a threshold, ARD$^+$ performs better than ARD, it demonstrates dropping irrelevant properties directly is a more effective strategy than assigning smaller weights on these items, which also verifies the effectiveness of our design to perform denoising in a hard-coding manner.

Compared with conventional and denoising-enhanced sequential models, the attribute-augmented model CARCA demonstrates superior performance. This illustrates the advantages of incorporating supplementary attributes to enhance the quality of recommendations. Fascinatingly, through the incorporation of additional properties into our model, ComeD$^+$ outperforms both CARCA and ComeD across the four datasets. This observation underscores the fact that even with the integration of supplementary attribute features, ComeD$^+$ consistently showcases remarkable denoising proficiency when dealing with diverse feature types, consequently yielding a superior performance. Finally, by considering a denoising process from a more microscopic perspective, both ComeD and ComeD$^+$ demonstrate exceptional performance across all four datasets.

5.3 Ablation Study

ComeD utilizes the neural bandit and a generated reward map for recommendation. To verify the effectiveness of such a design, we conduct the ablation study to analyze the contribution of each part.

5.3.1 Soft Attention vs. Hard Coding. Recall that ComeD formalizes the characteristic denoising task into a Contextual Multi-armed Bandit problem to perform denoising, here we first analyze whether such a hard-coding manner can bring benefit. Specifically, for each dimension of the item interacted in the sequence, we devise a new Multilayer Perceptron to replace each neural MAB of ComeD. The newly designed Multilayer Perceptron receives an input of the current sequential embedding concatenating with the corresponding dimension and outputs a probability revealing the significance of the dimension. By normalizing all outputs, we use them to weigh each feature and renovate the sequential embedding the same as the previous setting. We name the new model ComeD$_{soft}$. According to such a design, all features are preserved, and ComeD$_{soft}$ utilizes a soft attention mechanism to indicate the significance of each feature. Table 3 shows the performance comparison between ComeD and ComeD$_{soft}$.

Table 3: Performance of ComeD and ComeD$_{\text{soft}}$ over four datasets. The best performance is written in bold font.

Dataset	Metrics	ComeD$_{\text{soft}}$	ComeD
Beauty	Recall@5	17.13	**18.10**
	NDCG@5	12.65	**13.48**
	Recall@10	22.69	**23.52**
	NDCG@10	14.45	**15.24**
Sports	Recall@5	13.53	**14.24**
	NDCG@5	9.20	**9.56**
	Recall@10	18.53	**19.43**
	NDCG@10	11.25	**11.87**
Movies	Recall@5	19.79	**21.55**
	NDCG@5	14.03	**15.03**
	Recall@10	27.22	**28.83**
	NDCG@10	16.43	**17.51**
Tmall	Recall@5	24.36	**25.14**
	NDCG@5	19.31	**20.11**
	Recall@10	28.90	**29.95**
	NDCG@10	20.78	**21.47**

We can observe that ComeD is better than ComeD$_{\text{soft}}$. This demonstrates the effectiveness of performing a hard-coding mechanism for feature-level denoising. Similar observations can also be found in other denoising tasks [26, 36]. Besides, an interesting observation is that ComeD$_{\text{soft}}$ beats ARD on all datasets. The reason is that though both ARD and ComeD$_{\text{soft}}$ apply the soft attention mechanism for denoising, ARD concerns an aspect level of grouping several features, which is rougher than ComeD$_{\text{soft}}$ modeling each feature respectively. This further verifies our assumption, which is more convenient to perform a denoising task from feature level for a better recommendation.

*5.3.2 Feature-level Reward vs Item-level Reward vs. Sequence-level Reward.*ComeD divides the similarity between renovated sequential embedding and the target representation into a reward map to guide the learning of each bandit from feature level. To test its effectiveness, in this section, we propose several different map generation strategies and analyze their performance. In total, we consider two extra assignment mechanisms for comparison:

- Reward assignment from sequence level: The similarity score $s(\mathbf{l}_t, \mathbf{e}_{n+1})$ obtained in Eq. 4 is directly utilized to tune all actions of all bandits. By this, all bandits share the same observed reward. We name the new model as ComeD$_{\text{seq}}$.
- Reward assignment from item level: based on the learned similarity score obtained according to Eq. 4, we use the following function to compute the observed reward each action triggered:

$$\hat{r}_{a_t} = \alpha_i \cdot s(\mathbf{l}_t, \mathbf{e}_{n+1}); \alpha_i = \frac{exp(\mathbf{e}_i \cdot \mathbf{e}_{n+1})}{\sum_j exp(\mathbf{e}_j \cdot \mathbf{e}_{n+1})} \tag{7}$$

Compared with the reward assignment in Eq. 5, the designed mechanism only concerns the differences from item level, and the same dimension of items share the same reward, denoted by ComeD$_{\text{item}}$.

The results of ComeD and its two variants on four datasets are shown in Fig. 3. We see that ComeD$_{\text{seq}}$ obtains the worst performance on all datasets, demonstrating it is not an effective approach of assigning a single reward to guide the actions of all bandits, which ignores personal characteristics among different bandits.

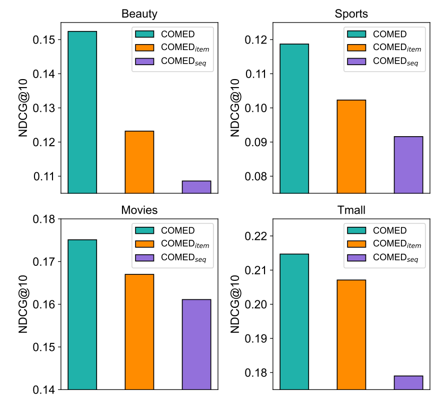

Figure 3: Performance comparisons when considering different reward assignment mechanisms on four datasets.

Comparing with ComeD$_{\text{seq}}$, ComeD$_{\text{item}}$ performs better by considering their differences from item level. undoubtedly, our ComeD analyzes interactions from a more granular standpoint, leading to the attainment of optimal performance.

5.4 Performance Analysis over Feature Count

In this section, we aim to analyze the performance when verifying the feature count for each bandit managed. specifically, we set the feature count within the scope of $\{1, 2, 4, 8, 16, 32, 64\}$ on four datasets, and report the test performance of ComeD in terms of NDCG@10 against the number of features over the four datasets in Fig. 4. Note that when the count is set to 64, ComeD assigns one bandit to manage all item dimensions, which turns to be an item-level denoising model, and within the scope of $\{2, 4, 8, 16, 32\}$, it becomes an aspect-aware model .

We can see that ComeD performs worst when the count is set to 64, it demonstrates that compared with the feature-level denoising mechanism, the item-level denoising approach is too rough to model the item relations well. The performance of ComeD increases when the count decreases on all datasets, verifying our assumption that it is necessary to make a denoising process from a fine-grained level.

5.5 Analysis on Swift Learning Strategy

ComeD designs a swift learning strategy to accelerate the corporation between sequential information and actions of bandits. In this

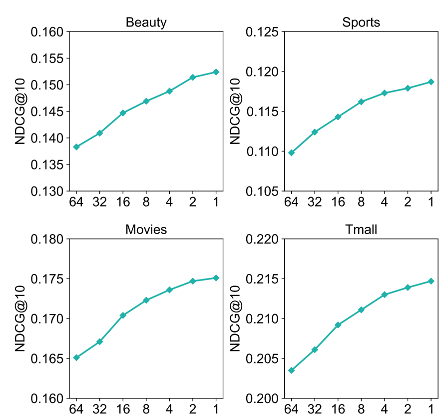

Figure 4: Performance variation in terms of NDCG@10 against the feature count within the scope of $\{1, 2, 4, 8, 16, 32, 64\}$.

Table 4: Performance comparison between ComeD and ComeD$_{serial}$ over four datasets.The best performance is in bold font.

Dataset	Metrics	ComeD$_1$	ComeD$_3$	ComeD$_5$	ComeD$_{serial}$
Beauty	Recall@5	17.76	18.10	18.18	**18.32**
	NDCG@5	13.21	13.48	13.58	**13.79**
	Recall@10	23.14	23.52	23.65	**23.90**
	NDCG@10	14.96	15.24	15.35	**15.53**
Sports	Recall@5	13.95	14.24	14.29	**14.59**
	NDCG@5	9.33	9.56	9.62	**9.98**
	Recall@10	19.09	19.43	19.52	**19.87**
	NDCG@10	11.47	11.87	12.02	**12.26**
Movies	Recall@5	21.13	21.55	21.70	**22.01**
	NDCG@5	14.80	15.03	15.11	**15.36**
	Recall@10	28.51	28.83	29.03	**29.25**
	NDCG@10	17.30	17.51	17.63	**17.82**
Tmall	Recall@5	24.72	25.14	25.30	**25.67**
	NDCG@5	19.95	20.11	20.32	**20.53**
	Recall@10	29.56	29.95	30.09	**30.31**
	NDCG@10	21.12	21.47	21.53	**21.73**

section, we further analyze its benefit from both the performance and efficiency perspectives. Specifically, we vary iteration steps for ComeD and name the variants as ComeD$_m$, where $m \in \{1, 3, 5\}$. Furthermore, we remove the acceleration strategy from ComeD and degenerate it into the previous serial operating approach. We name the new model as ComeD$_{serial}$. The performance of the variants of ComeD on four datasets are shown in Table 4.

We can observe that ComeD almost converge when $m \geq 3$. Besides, we can see that ComeD$_{serial}$ performs slightly better than ComeD$_3$ and ComeD$_5$. This demonstrates that the designed parallel metric to choose arms does not cause a server performance cost, showing the strong effectiveness of our acceleration strategy. Furthermore, owing to its sequential configuration, the computational runtime complexity of ComeD$_{serial}$ exhibits a linear relationship with the total count of features within an interacted sequence. While ComeD smashes this limit according to the parallel design. We take Beauty dataset as an example when performing models on a Linux PC with CPU (Intel(R) Xeon(R) Silver 4216, 16 cores, 32 threads, 2.10GHz) and GPU (Nvidia RTX 3090 24G), training ComeD takes about 0.5 hours, while ComeD$_{seiral}$ needs 15 hours to converge.

5.6 Further Analysis

To check the effectiveness of ComeD in performing feature-level denoising, in this section, we make a simple example and visualize it for further analysis. Specifically, we create two sequential patterns ***Vicky Cristina Barcelona*** $\rightarrow$ ***Titanic*** and ***Indentity*** $\rightarrow$ ***Sherlock Holmes*** that is nonexistent in the Amazon Movies dataset. By feeding it to ComeD, we map all generated representations into a 2-dimensional embedding space by PCA, and the visualization result of all representations are shown in Fig. 5.

We can see that according to a denoising process operated by ComeD, the movie ***Before Sunset*** is quite close to the renovated sequential embedding filtered from the designed pattern ***Vicky Cristina Barcelona→Titanic***. The underlying reason might be that the genre semantics (romantic) of two movies are derived from the movies, making the renovated sequential embedding aggregated by these two movies close to ***Before Sunset***, which is also a romantic movie. Interestingly, we find the denoised semantics

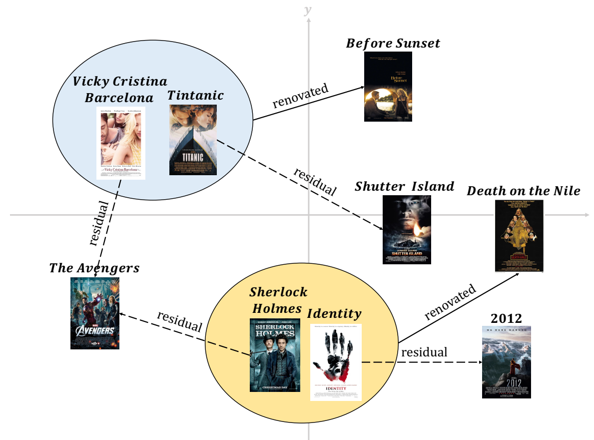

Figure 5: Visualization of sequential pattern *Vicky Cristina Barcelona*→Titanic and *Sherlock Holmes*→*Identity* on Amazon Movie dataset. The solid lines point to the position of the renovated sequential embedding, while the dashed lines point to the position of residual features discarded by ComeD.

from ***Vicky Cristina Barcelona*** is close to another movie led by ***Scarlett Johansson***(i.e. ***The Avengers***), while the denoised semantic from ***Titanic*** is relevant to ***Shutter Island***. It appears that actor attributes are absent in ComeD, which renders these two films similar to other movies the respective actors starred. Similar observations are also found in the other sequence pattern, where ComeD (***Indentity→Sherlock Holmes***) ≈ ***Death on the Nile***, ***Sherlock Holmes*** - *ComeD* (***Sherlock Holmes***, ***Identity***) ≈ ***The Avengers***, and ***Identity*** - *ComeD* (***Indentity→Sherlock Holmes***) ≈ ***2012***. It shows a superior ability to group relevant features for an effective recommendation. This verifies that ComeD is able to extract relevant features to fit complex item relations.

Besides, we also count the irrelevant features that our ComeD removed. Specifically, the average percent of removed features for each item are 51.2%, 53.6%,48.5%, and 50.1% on the testing sets of Beauty, Sports, Movies, and Tmall, respectively. By dropping these irrelevant features, ComeD obtains a flexible and effective recommendation on all datasets.

6 CONCLUSION

In this paper, we address an embedding denoising task in sequential recommendation scenario. The major novelty of ComeD lies in that it formalizes the representation denoising task as a Contextual Multi-armed Bandit problem. Specifically, ComeD assigns a two-armed neural bandit for each dimension of items, and a reward assignment mechanism is further designed to estimate the benefit that each action triggered. After that, a swift learning strategy is further introduced to leverage the actions of all bandits in parallel to facilitate the learning procedure. Experiments on four various datasets verify the effectiveness of our proposed model.

To our knowledge, it is the first time performing a feature-level denoising process in a hard-coding manner for sequential recommendation. In the future, we aim to design a general pre-training framework of performing feature denoising atomically to support various downstream tasks.

REFERENCES

[1] Djallel Bouneffouf and Irina Rish. 2019. A Survey on Practical Applications of Multi-Armed and Contextual Bandits. *CoRR* abs/1904.10040 (2019).

[2] Wei Cai, Weike Pan, Jingwen Mao, Zhechao Yu, and Congfu Xu. 2022. Aspect Re-distribution for Learning Better Item Embeddings in Sequential Recommendation. In *RecSys*. 49–58.

[3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *DLRS@RecSys*. 7–10.

[4] Zhiying Deng, Jianjun Li, Zhiqiang Guo, Wei Liu, Li Zou, and Guohui Li. 2023. Multi-view Multi-aspect Neural Networks for Next-basket Recommendation. In *SIGIR*. 1283–1292.

[5] Claudio Gentile, Shuai Li, and Giovanni Zappella. 2014. Online Clustering of Bandits. In *ICML*, Vol. 32. 757–765.

[6] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *IJCAI*. 1725–1731.

[7] Nicolas Gutowski, Tassadit Amghar, Olivier Camp, and Fabien Chhel. 2019. Gorthaur: A Portfolio Approach for Dynamic Selection of Multi-Armed Bandit Algorithms for Recommendation. In *ICTAI*. 1164–1171.

[8] Negar Hariri, Bamshad Mobasher, and Robin D. Burke. 2014. Context adaptation in interactive recommender systems. In *RecSys*. 41–48.

[9] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW*. 173–182.

[10] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *ICLR*.

[11] Cheng Hsu and Cheng-Te Li. 2021. RetaGNN: Relational Temporal Attentive Graph Neural Networks for Holistic Sequential Recommendation. In *WWW*. 2968–2979.

[12] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-Excitation Networks. In *CVPR*. 7132–7141.

[13] Tongwen Huang, Zhiqi Zhang, and Junlin Zhang. 2019. FiBiNET: combining feature importance and bilinear feature interaction for click-through rate prediction. In *RecSys*. 169–177.

[14] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *ICDM*. 197–206.

[15] Walid Krichene and Steffen Rendle. 2020. On Sampled Metrics for Item Recommendation. In *KDD*. 1748–1757.

[16] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *CIKM*. ACM, 1419–1428.

[17] Jiacheng Li, Yujie Wang, and Julian J. McAuley. 2020. Time Interval Aware Self-Attention for Sequential Recommendation. In *WSDM*. 322–330.

[18] Kaiyuan Li, Pengfei Wang, and Chenliang Li. 2022. Multi-Agent RL-based Information Selection Model for Sequential Recommendation. In *SIGIR*. 1622–1631.

[19] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. 2016. Collaborative Filtering Bandits. In *SIGIR*. 539–548.

[20] Zihan Lin, Hui Wang, Jingshu Mao, Wayne Xin Zhao, Cheng Wang, Peng Jiang, and Ji-Rong Wen. 2022. Feature-aware Diversified Re-ranking with Disentangled Representations for Relevant Recommendation. In *KDD*. 3327–3335.

[21] Zheng Liu, Jianxun Lian, Junhan Yang, Defu Lian, and Xing Xie. 2020. Octopus: Comprehensive and Elastic User Representation for the Generation of Recommendation Candidates. In *SIGIR*. 289–298.

[22] Chen Ma, Peng Kang, and Xue Liu. 2019. Hierarchical Gating Networks for Sequential Recommendation. In *KDD*. 825–833.

[23] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled Self-Supervision in Sequential Recommenders. In *KDD*. 483–491.

[24] Athanasios N. Nikolakopoulos, Dimitris Berberidis, George Karypis, and Georgios B. Giannakis. 2019. Personalized diffusions for top-n recommendation. In *RecSys*. 260–268.

[25] Vianney Perchet, Philippe Rigollet, Sylvain Chassang, and Erik Snowberg. 2015. Batched Bandit Problems. In *COLT*, Vol. 40. 1456.

[26] Yuqi Qin, Pengfei Wang, and Chenliang Li. 2021. The World is Binary: Contrastive Learning for Denoising Next Basket Recommendation. In *SIGIR*. 859–868.

[27] Ahmed Rashed, Shereen Elsayed, and Lars Schmidt-Thieme. 2022. Context and Attribute-Aware Sequential Recommendation via Cross-Attention. In *Recsys*. 71–80.

[28] Royi Ronen, Noam Koenigstein, Elad Ziklik, and Nir Nice. 2013. Selecting content-based features for collaborative filtering recommenders. In *RecSys*. 407–410.

[29] Chenglei Shen, Xiao Zhang, Wei Wei, and Jun Xu. 2023. HyperBandit: Contextual Bandit with Hypernewtork for Time-Varying User Preferences in Streaming Recommendation. In *CIKM*. ACM, 2239–2248.

[30] Qicai Shi, Feng Xiao, Douglas Pickard, Inga Chen, and Liang Chen. 2023. Deep Neural Network with LinUCB: A Contextual Bandit Approach for Personalized Recommendation. In *WWW*. 778–782.

[31] Linqi Song, Cem Tekin, and Mihaela van der Schaar. 2016. Online Learning in Large-Scale Contextual Recommender Systems. *IEEE Trans. Serv. Comput.* 9, 3 (2016), 433–445.

[32] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *CIKM*. 1441–1450.

[33] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *WSDM*. 565–573.

[34] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *WSDM*. 565–573.

[35] Yu Tian, Bofang Li, Si Chen, Xubin Li, Hongbo Deng, Jian Xu, Bo Zheng, Qian Wang, and Chenliang Li. 2023. Multi-Scenario Ranking with Adaptive Feature Learning. In *SIGIR*. ACM, 517–526.

[36] Xiaohai Tong, Pengfei Wang, Chenliang Li, Long Xia, and Shaozhang Niu. 2021. Pattern-enhanced Contrastive Policy Learning Network for Sequential Recommendation. In *IJCAI*. ijcai.org, 1593–1599.

[37] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. 2020. Next-item Recommendation with Sequential Hypergraphs. In *SIGIR*. ACM, 1101–1110.

[38] Pengfei Wang, Yu Fan, Long Xia, Wayne Xin Zhao, Shaozhang Niu, and Jimmy X. Huang. 2020. KERL: A Knowledge-Guided Reinforcement Learning Model for Sequential Recommendation. In *SIGIR*. 209–218.

[39] Qing Wang, Chunqiu Zeng, Wubai Zhou, Tao Li, S. S. Iyengar, Larisa Shwartz, and Genady Ya. Grabarnik. 2019. Online Interactive Collaborative Filtering Using Multi-Armed Bandit with Dependent Arms. *IEEE Trans. Knowl. Data Eng.* 31, 8 (2019), 1569–1580.

[40] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled Graph Collaborative Filtering. In *SIGIR*. ACM, 1001–1010.

[41] Yejing Wang, Zhaocheng Du, Xiangyu Zhao, Bo Chen, Huifeng Guo, Ruiming Tang, and Zhenhua Dong. 2023. Single-shot Feature Selection for Multi-task Recommendations. In *SIGIR*. 341–351.

[42] Yejing Wang, Zhaocheng Du, Xiangyu Zhao, Bo Chen, Huifeng Guo, Ruiming Tang, and Zhenhua Dong. 2023. Single-shot Feature Selection for Multi-task Recommendations. In *SIGIR*. ACM, 341–351.

[43] Qingyun Wu, Naveen Iyer, and Hongning Wang. 2018. Learning Contextual Bandits in a Non-stationary Environment. In *SIGIR*. 495–504.

[44] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-Based Recommendation with Graph Neural Networks. In *AAAI*. AAAI Press, 346–353.

[45] Xinxing Wu and Qiang Cheng. 2021. Fractal Autoencoders for Feature Selection. In *AAAI*. 10370–10378.

[46] Yanghong Wu, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Junhua Fang, and Fuzhen Zhuang. 2020. Vector-Level and Bit-Level Feature Adjusted Factorization Machine for Sparse Prediction. In *DASFAA*, Vol. 12112. 386–402.

[47] Xin Xin, Xiangnan He, Yongfeng Zhang, Yongdong Zhang, and Joemon M. Jose. 2019. Relational Collaborative Filtering: Modeling Multiple Item Relations for Recommendation. In *SIGIR*. 125–134.

[48] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph Contextualized Self-Attention Network for Session-based Recommendation. In *IJCAI*. 3940–3946.

[49] Pan Xu, Zheng Wen, Handong Zhao, and Quanquan Gu. 2022. Neural Contextual Bandits with Deep Representation and Shallow Exploration. In *ICLR*.

[50] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M. Jose, and Xiangnan He. 2019. A Simple Convolutional Generative Network for Next Item Recommendation. In *WSDM*. 582–590.

[51] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation. In *IJCAI*. 4320–4326.

[52] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. In *CIKM*. 4653–4664.

[53] Dongruo Zhou, Lihong Li, and Quanquan Gu. 2020. Neural Contextual Bandits with UCB-based Exploration. In *ICML*, Vol. 119. 11492–11502.

[54] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. In *CIKM*. 1893–1902.

[55] Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced MLP is All You Need for Sequential Recommendation. In *WWW*. 2388–2399.

[56] Yu Zhu, Jinghao Lin, Shibi He, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. 2020. Addressing the Item Cold-Start Problem by Attribute-Driven Active Learning. *IEEE Trans. Knowl. Data Eng.* 32, 4 (2020), 631–644.