User-Aware Multi-Interest Learning for Candidate Matching in Recommenders

Zheng Chai† Zhejiang University Hangzhou, China chaizheng.cz@gmail.com

Zhihong Chen Alibaba Group Hangzhou, China jhon.czh@alibaba-inc.com

Chenliang Li† Wuhan University Wuhan, China cllee@whu.edu.cn

Rong Xiao Alibaba Group Hangzhou, China xiaorong.xr@alibaba-inc.com

Houyi Li Alibaba Group Hangzhou, China lihouyi2008@126.com

Jiawei Wu Alibaba Group Hangzhou, China wujiawei@alibaba-inc.com

Jingxu Chen Alibaba Group Hangzhou, China xianlu@alibaba-inc.com

Haihong Tang Alibaba Group Hangzhou, China piaoxue@alibaba-inc.com

ABSTRACT

Recommender systems have become a fundamental service in most E-Commerce platforms, in which the matching stage aims to retrieve potentially relevant candidate items to users for further ranking. Recently, some efforts on extracting multi-interests from user's historical behaviors have demonstrated superior performance. However, the historical behaviors are not noise-free due to the possible misclicks or disturbances. Existing works mainly overlook the fact that the interests of a user are not only reflected by the historical behaviors, but also inherently regulated by the profile information. Hence, we are interested in exploiting the benefit of user profile in multi-interest learning to enhance candidate matching performance. To this end, a **u**ser-aware **m**ulti-**i**nterest learning framework (named UMI) is proposed in this paper to exploit both user profile and behavior information for candidate matching. Specifically, UMI consists of two main components: *dual-attention routing* and *interest refinement*. In the dual-attention routing, we firstly introduce a user-guided attention network to identify the important historical items with respect to the user profile. Then, the resultant importance weights are leveraged via the dual-attentive capsule network to extract the user's multi-interests. Afterwards, the extracted interests are utilized to highlight the corresponding user profile features for interest refinement, such that different user profiles can be incorporated into interest learning for diverse user preference understanding. Besides, to improve the model's discriminative capacity, we further devise a *harder-negatives* strategy to support model optimization. Extensive experiments show that UMI significantly outperforms state-of-the-art multi-interest modeling alternatives. Currently, UMI has been successfully deployed at Taobao App in Alibaba, serving hundreds of millions of users.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Candidate Matching, Multi-Interest Learning, Recommendation

ACM Reference Format:

Zheng Chai, Zhihong Chen, Chenliang Li, Rong Xiao, Houyi Li, Jiawei Wu, Jingxu Chen, Haihong Tang. 2022. User-Aware Multi-Interest Learning for Candidate Matching in Recommenders. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3477495.3532073

1 INTRODUCTION

Large-scale recommender systems have demonstrated great values in improving customer experience and boosting profits for many E-Commerce platforms. Due to the astronomical amount of items in E-Commerce platforms like Taobao, the industrial recommender systems [7, 21] generally consist of two stages: candidate *matching* and *ranking*. In the matching stage, thousands of items that are potentially relevant to the user interests are retrieved from the large-scale item pool, after which the ranking stage is performed to predict the likelihood that a user would interact with the item. As the fundamental part of the recommender system, the candidate matching has drawn increasing attention in recent years [3, 16, 23].

Various approaches have been developed for candidate matching in recommenders [25, 28]. Specifically, deep learning based methods, which generally represent users and items with low-dimensional vectors through a dual-tower architecture, have become the de facto standard practice for real-world scenarios [10]. For example, the Youtube DNN [3] transforms the user historical behaviors into a fixed-length dense vector representation. Then, the inner product

† Corresponding authors.

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3532073

between the user and item representations is calculated for estimating the relevance scoring for candidate matching. However, the representation learned through the dual-tower architecture is generally limited to characterize diverse interests of users, and thus becomes a bottleneck in enhancing the matching performance.

Recently, modeling the diverse user interests has shown promising performance gain for candidate matching. The multi-interest network with dynamic routing (MIND) [14] is the first attempt in representing a user with multiple interest vectors via deep neural network structures. MIND utilizes the capsule network (CapsNet) [24] to extract user interests, and a label-aware attention mechanism is performed for estimating the relevance to the target item. To enhance the diversity of the retrieved top-N items, a multi-interest framework for recommendation (ComiRec) [1] is devised by integrating a controllable yet flexible item aggregation module.

Despite significant performance gain obtained by these solutions, they mainly overlook the exploration of user profile information. Generally, a user would interact with amounts of items in the E-Commerce platform with little restriction. As many items are interacted because of misclicks or disturbances, it is essential to filter the noisy interacted items and capture the multiple interests that reflect the dominating preferences of a user. Note that the interests of a user are inherently self-possessed by users, which are also highly relevant to the user profile information. Thus, it is expected that the user profile information can enrich the semantic signals and achieve noise immunity. Besides, the fine-grained user interests can be deduced from his or her personal profiles. For example, a specific interest in dress would be highly correlated with the gender of the user, while the features like user's province information or phone operating system are not that relevant instead. Hence, various user profiles are of importance to be incorporated in deriving multiple interests for better user modeling.

Motivated by these problems, in this paper, a novel **u**ser-aware **m**ulti-**i**nterest learning framework (UMI) is proposed for candidate matching. Specifically, UMI consists of two main components: *dual-attention routing* (DAR) and *interest refinement* (IR). Firstly, to incorporate the user's personal information into the multi-interest modeling, a user-guided attention network (UGANet) is devised in DAR by taking both user profile and the interacted items as input. The resultant attention weights indicate the relevance to the user's interests and are leveraged via a dual-attentive dynamic routing mechanism for interest extraction. Afterwards, to obtain more personalized user interests which encode user profile information, the extracted interests are further utilized to identify relevant user features in the IR module. The aim is to absorb the relevant user profiles to refine different user interests, leading to better user preferences understanding. Besides, considering the current gap between the training and inference stages under the multi-interest learning framework, we further devise a *harder-negatives* (HN) strategy, in which the most relevant user interest is selected for each negative sample. This encourages the negatives to play a harder impact in the learning process, yielding enhanced discriminative capacity. Offline experiments on two real-world datasets, including a public benchmark and a large-scale industrial dataset from Taobao App, and online experiments on Taobao verify the effectiveness of our proposed UMI. Overall, the contribution of our work can be summarized as follows:

- We propose UMI for multi-interest learning in recommenders, which is the first attempt to comprehensively leverage both user profile information and user behaviors for multi-interest understanding. For this purpose, a DAR module and an IR module are developed, which identify the important historical items for multiple interests and fuse relevant user profile information for precise and diverse candidate matching.
- To enhance the discriminative capacity of the multi-interest framework, we devise a harder-negatives strategy to facilitate the model optimization, which aligns the training and inference stages under the multi-interest learning framework and significantly improves the recommendation performance. This strategy can be readily integrated with the current multi-interest modeling frameworks.
- Extensive experiments on two real-world datasets validate the effectiveness of the proposed method. Currently, UMI has been deployed at Taobao App, serving hundreds of millions of users and yielding significant improvement on a series of commercial metrics.

2 RELATED WORK

2.1 Deep Candidate Matching

The candidate matching serves as the fundamental step in industrial recommender systems, which retrieves a small subset (generally thousands) from a large corpus. Due to the success of deep learning in recent years, many efforts have been made for building neural matching models in recommenders. Neural CF constructs neural networks to model user and item interactions for collaborative filtering [8]. Two-tower-based DNNs have been widely developed in industrial recommenders, which independently model the user and item representations to guarantee computation efficiency in practice [3, 11]. The online deep matching can then be regarded as a searching process of the nearest neighbors from the item pool based on the dense-vector-formulated user representation. Besides, TDM [33] and JTM [32] provide novel viewpoints to capture user-item interactions via tree-based structures.

2.2 User Interest Modeling

Precisely modeling user interests is a crucial task in recommendation, while most current methods observe user interests from his or her behaviors only [5, 22, 27]. For example, deep interest network (DIN) [31] extracts user interest with regard to the candidate item from user behaviors, and the target attention mechanism is computationally expensive which makes DIN only suitable for the ranking stage. Furthermore, a deep interest evolution network [30] is developed which adopts the GRU unit for learning interest evolving trajectory in click-through rate prediction. Besides, GRU4Rec [9] and BST[2] adopted GRU or Transformer to elaborately learn the sequential relationship underlying user behaviors.

To capture user's multiple aspects of interests in the matching stage, the capsule network [24] has been widely adopted to generate multiple interest clusters in recommenders [14, 20]. Furthermore, to balance the recommendation accuracy and diversity, ComiRec [1] further integrates a controllable balance value function with a greedy inference algorithm for item aggregation. The sparse interest network [26] generates multiple interest embeddings from a large pool of intention prototypes. Another deep multi-interest network [27] is proposed which models user's multi-interest for the click-through rate prediction task in E-commerce platforms.

Despite the popularity, the existing solutions extract multiple interests from the user behaviors only, after which the static user profile is concatenated to output the final user representations. Differently, our proposed DAR integrates both user profile and behaviors for better characterizing user interests, and various user profiles are then leveraged for further interest refinement. In comparison with the attentive CapsNet in language modeling [29], as the attention from the target item [31] is computationally expensive in the matching stage, we leverage user profile information and design a UGANet for revealing the importance of interacted items.

3 METHODOLOGY

3.1 Problem Formulation

Let $\mathcal{U}$ denotes the set of users and $\mathcal{I}$ denotes the large-scale item pool. For user $u \in \mathcal{U}$, there are two kinds of features: the user profile $\boldsymbol{p}_u$, and the user behaviors $\mathcal{B}_u$ which represents the historical interacted items of user u: $(x_1, \cdots, x_T)$, where x_i is the ith item interacted by the user and T is the historical behavior length. The core task for candidate matching in industrial recommender systems is to retrieve a subset from $\mathcal{I}$, such that the retrieved ones are relevant to the user interests. Generally, the task in multi-interest based candidate matching can be formulated as follows:

$$r_{u,t} = f(\mathbf{O}_u, \boldsymbol{e}_t) = \max_{1 \leq k \leq K} \left((\boldsymbol{o}_u^k)^{\mathrm{T}} \boldsymbol{e}_t \right) \tag{1}$$

where $\mathbf{O}_u = \left(\boldsymbol{o}_u^1, \boldsymbol{o}_u^2, ..., \boldsymbol{o}_u^K \right) \in \mathbb{R}^{d_i \times K}$ denotes the K interest vectors for user u and $\boldsymbol{e}_t \in \mathbb{R}^{d_i}$ denotes the representation vector for target item x_t. $r_{u,t}$ is the relevance score between u and each target item $x_t \in \mathcal{I}$, in terms of which the top N candidate items $\mathcal{I}_N$ can be retrieved for the user as the recommendation results. Due to the huge amount of possible items in E-Commerce platforms, the fast nearest neighbors (*e.g.,* Faiss [12]) is generally adopted to generate the candidate items for top-N matching. Similar to the ranking stage, the evaluation for the matching stage also examines the capacity of successfully retrieving the items that will be interacted by the user.

3.2 Feature Composition and Embedding Layer

An overview of UMI is illustrated in Figure 1. As mentioned above, three kinds of inputs are considered in our recommender system: user profile, user historical behaviors, and target item. The user profile includes many personal features like gender, province, mobile operating system, career, married or not, etc. Besides, each item in the user behaviors has the same feature fields as the target item, including item id, item category, brand, and so on. Among these user and item feature fields, both one-hot features and continuous features are included. As to the continuous features, we first discretize them into one-hot vectors. Then, an embedding layer is conducted to project the one-hot features into fixed-size dense embeddings. After this, features within a field are concatenated into a field representation. We then form the user profile and item embeddings by concatenating the embeddings of the corresponding fields together. For clarity, we denote the embeddings of the user profile, the ith item x_i from the historical behaviors, and the target item as $\boldsymbol{e}_u \in \mathbb{R}^{d_u}$, $\boldsymbol{e}_i \in \mathbb{R}^{d_i}$, and $\boldsymbol{e}_t \in \mathbb{R}^{d_i}$, respectively. For example, given C different fields in the user profile, $\boldsymbol{e}_u$ can be represented as $\boldsymbol{e}_u = [\boldsymbol{e}_u^{(1)}, \boldsymbol{e}_u^{(2)}, \cdots, \boldsymbol{e}_u^{(C)}]$, where $\boldsymbol{e}_u^{(c)}$ is the embedding of the cth field.

3.3 Dual-Attention Routing (DAR)

After the embedding layer, we first utilize the proposed DAR module to extract the multiple interests by taking both user profile and user behaviors into account.

User-Guided Attention Network. Normally, due to the randomness and diversity of a user's behaviors, it is of significance to pay different attention to those historical items, such that the real interests can be extracted precisely. Given the embedding vector $\boldsymbol{e}_i$ of the ith historical item x_i and the user profile $\boldsymbol{e}_u$, the two-layer MLP is performed to calculate the attention weight of item x_i as follows:

$$a_i = sigmoid(\mathbf{W}_2^{\mathrm{T}} \mathrm{ReLU}(\mathbf{W}_1^{\mathrm{T}} [\boldsymbol{e}_i, \boldsymbol{e}_u] + \boldsymbol{b}_1) + b_2) \tag{2}$$

where $\mathbf{W}_1 \in \mathbb{R}^{(d_i + d_u) \times d_h}$, $\mathbf{W}_2 \in \mathbb{R}^{d_h}$, $\boldsymbol{b}_1 \in \mathbb{R}^{d_h}$ and b_2 are trainable parameters, d_h is the number of hidden dimensions, $[\cdot, \cdot]$ denotes the concatenation operator of two vectors.

Note that different from conventional self-attentive method [18] that is widely used in behavior sequence aggregation in recommender systems [1, 26], our UGANet incorporates the user profile information, which explicitly helps identify the importance of item x_i with respect to the user's real interests.

Multi-Interest Extraction. Capsule network has been proved to be effective in digital recognition [4, 24], relation extraction [19, 29], and interest clustering in recommender systems [14, 15, 20]. In CapsNet, a capsule works as a cluster such that the input vectors related to a specific concept are aggregated together. Specifically, we utilize the dynamic routing mechanism to generate multiple interests for the user. That is, each capsule is considered as an interest aggregator (namely *interest capsule*). However, in recommender systems, we argue that the user's interests are not a simple reflection of his or her behaviors, but also regulated by the user profile information. To precisely characterize user's interests, we devise a DAR algorithm that incorporates the importance of each historical item a_i revealed by user profiles into the dynamic routing process.

Given the embeddings of these historical items, $\mathbf{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_T]$, the jth interest capsule calculates the interest vecotr $\boldsymbol{z}_j \in \mathbb{R}^{d_i}$ as follows:

$$\boldsymbol{z}_j = \sum_{i=1}^{T} a_i \boldsymbol{c}_{ij} \mathbf{W}_j \boldsymbol{e}_i \tag{3}$$

where $\mathbf{W}_j \in \mathbb{R}^{d_i \times d_i}$ is the transformation matrix for the jth interest capsule, a_i denotes the user's attention weight on ith historical item, and $\boldsymbol{c}_{ij}$ is the clustering probability for item x_i under the jth interest capsule. Similar to the dynamic routing mechanism proposed for CapsNet, the clustering probability is calculated as follows:

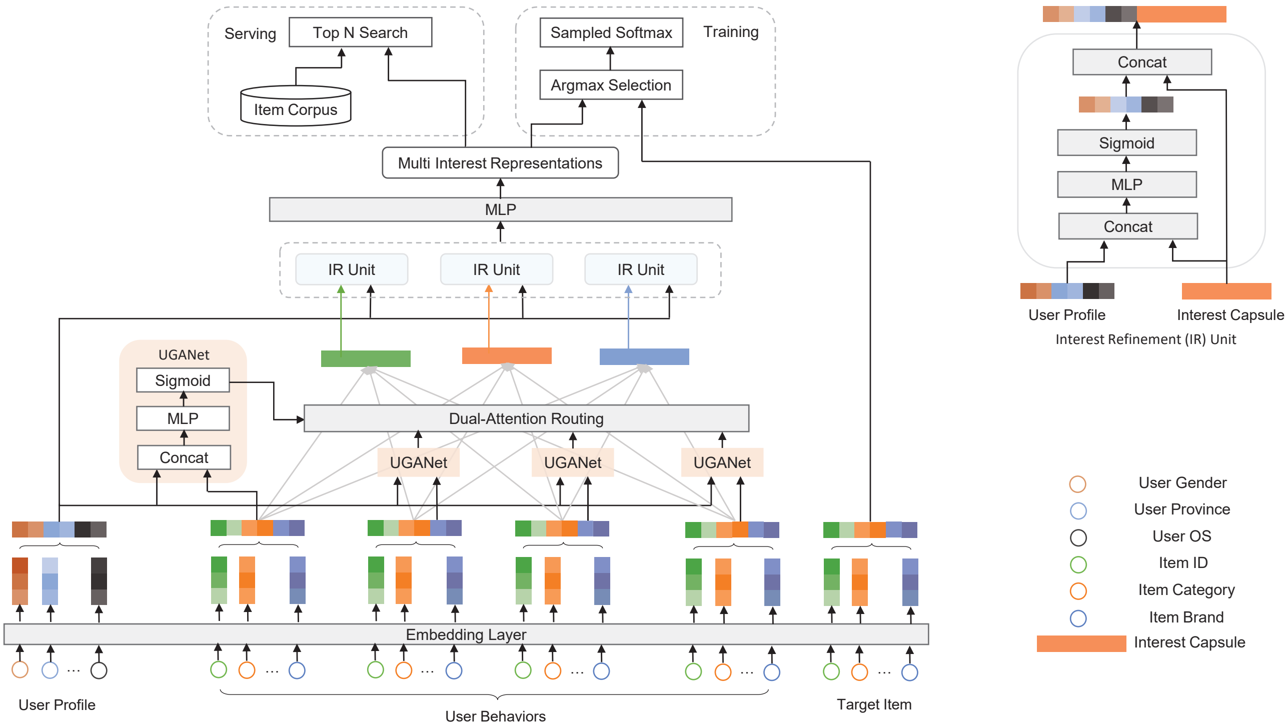


Figure 1: Overview of UMI. Given user profile and behaviors as input, UGANet first learns attention weights and reveals user's intentions on the interacted items. The DAR module then extracts multiple user interests based on both the UGANet weights and the user historical behaviors. Then, each interest is refined with the IR module by a two-layer MLP with ReLU activation (*Best viewed in color*).

$$c_{ij} = \frac{\exp(\boldsymbol{b}_{ij})}{\sum_{k=1}^{K} \exp(\boldsymbol{b}_{ik})} \tag{4}$$

where $\boldsymbol{b}_{ij}$ denotes the coupling coefficients, a.k.a. the routing logits, between the transformed item embedding $\mathbf{W}_j \boldsymbol{e}_i$ and the "squashed" interest vector $\boldsymbol{v}_j$:

$$\boldsymbol{b}_{ij} = (\mathbf{W}_j \boldsymbol{e}_i)^{\mathrm{T}} \boldsymbol{v}_j \tag{5}$$

$$\boldsymbol{v}_j = \mathrm{squash}(\boldsymbol{z}_j) = \frac{\|\boldsymbol{z}_j\|^2}{\|\boldsymbol{z}_j\|^2 + 1} \frac{\boldsymbol{z}_j}{\|\boldsymbol{z}_j\|} \tag{6}$$

where $\|\cdot\|^2$ denotes the length of a vector.

Given that the calculation of the routing logits relies on both $\boldsymbol{e}_i$ and $\boldsymbol{v}_j$ while the value of $\boldsymbol{v}_j$ also depends on the routing logits, a dynamic routing process is designed to iteratively update these two values, in which the logits $\boldsymbol{b}_{ij}$ are firstly initialized with Gaussian distribution. As both the user's attention on the ith historical items, *i.e.*, a_i, and the ith item's attention on K interest capsules, *i.e.*, $\boldsymbol{c}_{ij}$, are considered in our process, this routing algorithm is termed as *dual-attention routing*. The detail of this routing process is described in Algorithm 1. Here, the iterative number t is set to 3, which is also the default setting in CapsNet [24]. The updated $\boldsymbol{v}_j$ after the last iteration is then considered as the jth interest.

Algorithm 1 Dual-Attention Routing Algorithm

Input: User profile embedding $\boldsymbol{e}_u$, user behavior embeddings $\mathrm{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_{\mathrm{T}}]$, iterative number t, number of interest capsules K, standard deviation σ for random initialization of $\boldsymbol{b}_{ij}$
Output: User interest capsules $\mathbf{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_{\mathrm{K}}]$

```
for all e_i and v_j pair do
    Initialize b_ij with random value from N(0, σ^2)
end for
Calculate the attention weights a_i with Equation 2
for t iterations do
    c_ij = softmax(b_ij), ∀e_i ∈ E
    v_j = squash(∑_i a_i c_ij W_j e_i), ∀v_j ∈ V
    b_ij = (W_j e_i)^T v_j, ∀v_j ∈ V and e_i ∈ E
end for
```

3.4 Interest Refinement (IR)

With K interests extracted above, we further leverage the user profile information to refine each of them respectively. As different interests can be regarded as reflections of different aspects of user's personal profile, each interest vector is firstly leveraged to identify the relevant user features for the following refinement. Recall that we form $\boldsymbol{e}_u$ with C different field embeddings, $\boldsymbol{e}_u =$

$[\boldsymbol{e}_u^{(1)}, \boldsymbol{e}_u^{(2)}, \cdots, \boldsymbol{e}_u^{(C)}]$. The relevance weights for these features are then calculated as follows:

$$\boldsymbol{d}_k = f(\boldsymbol{v}_k, \boldsymbol{e}_u^{(1)}, \boldsymbol{e}_u^{(2)}, \cdots, \boldsymbol{e}_u^{(C)}) \tag{7}$$

where $\boldsymbol{d}_k \in \mathbb{R}^C$ is the attentions weights on the different features in the user profile $\boldsymbol{e}_u$. Similar to the UGANet, $f(\cdot)$ concatenates the input vectors and is formulated as a two-layer ReLU-MLP with a sigmoid output activation function.

Based on the weights $\boldsymbol{d}_k$, the interest-wise user profile embedding can be reformulated as:

$$\boldsymbol{e}_u^k = \text{Concat}\left(\boldsymbol{d}_{k1}(\boldsymbol{e}_u^{(1)}), \boldsymbol{d}_{k2}(\boldsymbol{e}_u^{(2)}), ..., \boldsymbol{d}_{kC}(\boldsymbol{e}_u^{(C)})\right) \tag{8}$$

where $\boldsymbol{d}_{kc}$ denotes the cth element of $\boldsymbol{d}_k$. It is intuitive that these relevant profile features could help us refine the corresponding user interest for better preference learning. Hence, we derive the final kth user interest $\boldsymbol{o}_u^k$ through another two-layer MLP with ReLU as follows:

$$\boldsymbol{o}_u^k = \text{MLP}([\boldsymbol{e}_u^k, \boldsymbol{v}_k]), k = 1, 2, ..., K \tag{9}$$

Here, we can see that the relevant user profile $\boldsymbol{e}_u^k$ is injected to refine the corresponding interest vector, resulting in more personalized and diverse interests learning of $\boldsymbol{o}_u^k$.

3.5 Training and Model Deployment

With the multiple user interests $\mathbf{O}_u$ and the target item representation $\boldsymbol{e}_t$, an argmax operator is adopted to select the user interest that is most relevant to the target item. This can be regarded as a process that the target item x_t activates a specific interest from the user interest pool $\mathbf{O}_u$:

$$\boldsymbol{o}_u = \mathbf{O}_u[\text{argmax}(\mathbf{O}_u^{\mathrm{T}}\boldsymbol{e}_t)] \tag{10}$$

Based on the selected user interest, the likelihood that user u will interact with item x_t can be calculated as follows:

$$p(x_t|u) = \frac{\exp(\boldsymbol{o}_u^{\mathrm{T}}\boldsymbol{e}_t)}{\sum_{x_j \in \mathcal{I}} \exp(\boldsymbol{o}_u^{\mathrm{T}}\boldsymbol{e}_j)} \tag{11}$$

The model training is then to maximize this likelihood score for each positive target item in the training set against the rest negative ones.

Harder-Negatives (HN). The above training paradigm has become a common practice in multi-interest framework [1, 14]. However, when observed carefully we can find there is a significant gap between the numerator and denominator in Equation 11. The relevance between the negative item x_j and the user is calculated through the *negative* item representation $\boldsymbol{e}_j$ and the user interest $\boldsymbol{o}_u$ that is most relevant to the *positive* item x_t. This would cause a significant gap between the training and the inference under the multi-interest framework, as the $\boldsymbol{o}_u$ selected by $\boldsymbol{e}_t$ is remarkably irrelevant with the negatives $\boldsymbol{e}_j$, where $x_j \in \mathcal{I}$ and $x_j \neq x_t$. To fulfill this gap, we construct harder-negatives (HN), which can enhance the impact of the negatives and be readily incorporated into the multi-interest learning framework. Specifically, the argmax operator is used by each item including both the positive and the negatives:

$$\boldsymbol{o}_u^{(q)} = \mathbf{O}_u[\text{argmax}(\mathbf{O}_u^{\mathrm{T}}\boldsymbol{e}_q)] \tag{12}$$

where q can be the target item x_t, *i.e.,* the positive item for training, or the negatives $x_j \in \mathcal{I}$ where $x_j \neq x_t$. The operator in Equation 12 can be regarded as an activation process from the user representation pool $\mathbf{O}_u$, where the query item can be both the positive or the negative ones. Accordingly, the likelihood in Equation 11 can be reformulated as:

$$p(x_t|u) = \frac{\exp\left((\boldsymbol{o}_u^{(t)})^{\mathrm{T}}\boldsymbol{e}_t\right)}{\sum_{x_j \in \mathcal{I}} \exp\left((\boldsymbol{o}_u^{(j)})^{\mathrm{T}}\boldsymbol{e}_j\right)} \tag{13}$$

The objective function of the UMI is to minimize the following negative log-likelihood:

$$\mathcal{L} = \sum_{u \in \mathcal{U}} \sum_{x_t \in \mathcal{I}_u} -\log p(x_t|u) \tag{14}$$

where $\mathcal{I}_u$ denotes the historical items set of user u. Due to the tremendous size of the corpus $\mathcal{I}$, the sampled softmax [3] is adopted to make the training tractable.

After UMI is trained, the overall network structure can be deployed for online serving. Specifically, the item field features are used once to generate the item embeddings which are saved as item corpus. Then, the user profile and the user historical behavior embeddings are fed into UMI to generate multiple interest vectors for each user. As shown in Figure 1, we implement online real-time top-N matching based on the fast nearest neighbors approach [12]. These results are considered as the candidates for the later ranking stage in recommender systems.

3.6 Connections with Existing Methods

DIN. Our proposed DAR is inspired by DIN [31], which identifies the important historical items through the attention mechanism. Differently, the items are weighted by the user profile information in DAR rather than the target item in DIN, since the number of possible candidate items considered by the matching stage is several orders of magnitude greater than the ranking stage. That is, our proposed UMI is more cost-effective for the matching stage.

MIND and ComiRec. MIND [14] and ComiRec [1] are both state-of-the-art multi-interest modeling solutions, which adopt CapsNet to generate different interest vectors and treat each item in the user behavior sequence equally. On the contrary, we argue that different items play nonequal importance for revealing various user preferences, and various instead of static user profiles are of significance to be incorporated for diverse interest learning. Besides, an HN strategy is developed, which makes the training and inference stages in the multi-interest framework more consistent, and thus facilitates the discriminative ability of the model.

Table 1: Statistics of the two datasets.

Dataset	#Users	#Items	#Interaction
MovieLens	6,040	3,260	998,539
Alibaba	590,958,088	1,910,452	225,396,818

4 EXPERIMENTS

In this section, we first compare the proposed UMI with the conventional and multi-interest solutions on two real-world datasets, and a comprehensive ablation study is performed to verify the DAR, IR, and HN modules respectively. Then, the online A/B test results are reported by deploying UMI on the recommender system of Taobao App in Alibaba. Finally, further analysis and case study is conducted to illustrate the effectiveness of UMI.

4.1 Offline Experiments

Datasets and Experimental Setup. We conduct offline experiments on two real-world datasets: MovieLens[1] and an industrial dataset collected from Taobao App. The statistics of the two datasets are summarized in Table 1.

The movie rating dataset contains $6,040$ users, $3,260$ items, and $998,539$ interactions. The user profile features include user gender, age, and occupation information. For performance evaluation, we conduct experiments under a strong generalization setting [1, 17]. Specifically, all users are split into training, validation, and test data at the ratio of 80%, 10%, and 10%. For model training, a random item from the user interacted items is selected as the target item, and the sequence before the target item is selected as user historical behaviors. For evaluation on the validation and test set, the former 90% interacted items are used as the historical behaviors, and the rest 10% are selected as the target ones. The behavior sequences are truncated at the length of 10 (*i.e.,* $T = 10$), which is a common and reasonable practice for balancing complexity and accuracy.

The Alibaba dataset is collected from Taobao App, in which we use traffic logs of six weeks for training, and the samples in the following two days are used for validation and testing, respectively. There are around six hundred million users and two million items, resulting in two hundred million interactions between them. The historical behaviors are truncated at the length of 50 (*i.e.,* $T = 50$). This dataset has 89 primary and 5831 secondary categories. There are 37 features included in the user profiles, including province, gender, bc_prefer, OS, career, education degree, has pet/car/house or not, married or not, and so on. The feature "bc_prefer" indicates whether the user prefers to purchase items in the business-to-consumer (B2C) channel or the consumer-to-consumer (C2C) channel, both of which can be readily reached in the E-commerce platforms of Alibaba. Also, the features like item id, category, brand, and price level are included in the item side. The rich features in this large-scale data help UMI better recognize the items that users are really interested in, and thus provide a suitable platform for evaluating different candidate matching methods.

Comparing Methods. The proposed UMI is compared with the following representative methods for candidate matching:

[1] https://grouplens.org/datasets/movielens/1m/

- **DSSM.** DSSM [11] is devised to learn embeddings for users and items respectively in terms of the user profile and item profile information.
- **Youtube DNN.** Youtube DNN [3] adopts a mean-pooling operation over the embeddings of the user historical items, after which the user profile embedding is concatenated. Then, an MLP layer is utilized to derive the final user representation.
- **MIND.** MIND [14] uses a capsule network-based multi-interest extractor to cluster user behaviors, and the same user profile vector is concatenated with various capsules. Then an MLP layer is utilized to derive the multiple interests of a user.
- **ComiRec.** Compared with MIND, ComiRec [1] follows the vanilla dynamic routing algorithm used in CapsNet with a controllable and flexible item aggregation module.
- **MIND+HN and ComiRec+HN.** As our proposed HN strategy is a general training strategy for the multi-interest learning framework, we integrate the HN strategy into both the MIND and ComiRec models, and term them as MIND+HN and ComiRec+HN respectively.

Implementation Details. All these deep learning methods are implemented in Tensorflow[2] with an Adam optimizer [13]. All the hyper-parameters are determined through the validation set for a fair comparison. In MovieLens dataset, the number of hidden units for $\boldsymbol{o}_u^k$ in Equation 9 is 32, and this value is set to 8 for Equation 2 and 7. The batch size, learning rate, number of sampled negatives, and number of interests is set to 128, 0.005, 5, and 3 respectively. In Alibaba dataset, due to the larger-scale training set, the number of hidden units in the user tower is 128, and this value is set to 64 in Equation 2 and 7. The batch size is set to 1024 and the learning rate is 0.001. The number of sampled negatives is $10,240$, and the number of interest is set to 6.

Evaluation Metrics. For offline evaluation, we use **HitRate@N** and **Precision@N** metrics, which have been widely adopted for candidate matching:

$$\text{HitRate@}N = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{|\mathcal{P}_u \cap \mathcal{I}_u|}{|\mathcal{I}_u|} \tag{15}$$

$$\text{Precision@}N = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{|\mathcal{P}_u \cap \mathcal{I}_u|}{N} \tag{16}$$

where $\mathcal{P}_u$ denotes the top-N retrieved items for the user from the entire item pool.

4.2 Results and Discussion

Overall Performance. Table 2 shows the experimental results in terms of HitRate@N and Precision@N ($N = 10, 50, 300$) on the two datasets. Here, we can make several observations.

First, DSSM shows the worst performance among these methods, suggesting that the user behavior information is of great significance in user interest learning. Then, in comparison with the vanilla MIND and ComiRec, Youtube DNN shows strong ability in candidate matching. This may be mainly attributed to the fact that the vanilla MIND and ComiRec directly use the user representation

[2] https://www.tensorflow.org/

Table 2: Comparison results of different methods on the two datasets. The best and second best results are highlighted in boldface and underlined respectively. ▲% denotes the relative improvement of UMI over the best comparing method, and the improvements are statistically significant ($p < 0.05$). The results are percentage numbers with "%" omitted.

	MovieLens						Alibaba					
	Metrics@10		Metrics@50		Metrics@300		Metrics@10		Metrics@50		Metrics@300	
	HitRate	Precision	HitRate	Precision	HitRate	Precision	HitRate	Precision	HitRate	Precision	HitRate	Precision
DSSM	2.706	2.450	11.006	2.182	40.552	1.644	0.392	0.140	1.277	0.091	4.517	0.054
Youtube DNN	7.164	<u>6.887</u>	<u>26.687</u>	<u>5.417</u>	62.451	<u>2.571</u>	4.659	1.668	11.827	0.847	27.180	0.324
MIND	6.210	6.060	22.415	4.623	59.440	2.385	4.262	1.526	11.203	0.802	26.799	0.320
+HN	7.236	6.407	25.781	5.123	62.722	2.519	<u>4.981</u>	<u>1.783</u>	<u>12.780</u>	<u>0.915</u>	<u>29.620</u>	<u>0.354</u>
ComiRec	6.117	5.778	20.839	4.265	56.024	2.289	4.186	1.499	10.934	0.786	26.243	0.313
+HN	<u>7.270</u>	6.358	26.367	5.195	<u>63.012</u>	2.534	4.949	1.772	12.650	0.906	29.181	0.249
UMI	**7.505**	**7.036**	**27.805**	**5.649**	**64.820**	**2.647**	**5.454**	**1.953**	**13.605**	**0.974**	**30.628**	**0.366**
▲%	3.23	2.16	4.19	4.28	2.87	2.96	9.50	9.53	6.46	6.45	3.40	3.39

Table 3: Ablation study of the proposed UMI. The results are percentage numbers with "%" ommited.

Methods	Metrics@10		Metrics@50		Metrics@300	
	HitRate	Precision	HitRate	Precision	HitRate	Precision
UMI	**5.454**	**1.953**	**13.605**	**0.974**	**30.628**	**0.366**
UMI w/o DAR	4.235	1.516	11.197	0.802	26.909	0.321
UMI w/o IR	5.279	1.890	13.310	0.953	30.189	0.360
UMI w/o HN	5.172	1.852	13.001	0.931	29.667	0.354

selected by the target item for those negatives, which makes the negatives too "easy" to be recognized during the model training.

Second, when we incorporate the designed HN strategy into MIND and ComiRec, it can be observed that their performances are significantly improved. Moreover, both MIND+HN and ComiRec+HN readily outperform the single-interest methods including DSSM and Youtube DNN in many settings, especially in the more complex Alibaba dataset. This is mainly attributed to the consistent argmax selection for both positive and negative items in HN, which makes the model learn stronger discriminative capacity.

Finally, UMI achieves the best performance across the two datasets and different metrics. We believe that the reason is threefold: 1) The DAR exploits the user profile information to guide the interest extraction, leading to better noise immunity. 2) The relevant user profile information is further leveraged to refine the user interests. 3) The designed HN strategy further pushes UMI to reach more discriminative capacity in the multi-interest modeling framework.

Ablation Study. To further investigate the impact of different components (*i.e.,* DAR, IR, and HN) in our proposed UMI, we conduct the ablation study on the more challenging Alibaba dataset. To verify each component's impact, we disable one component each time and keep the other parts unchanged. Table 3 reports the performance of these variants. According to the table, it is obvious that UMI obtains the best performance across different metrics, confirming the validity of these components.

4.3 Online Experiments

Besides the offline experiments, we conduct online A/B test by deploying UMI in the recommender system of Taobao for a week. In the control group, the matching strategy deployed in our current production system is taken as the baseline. In the test group, the proposed UMI is deployed to serve the candidate matching. The same ranking strategy is used for a fair comparison. For evaluation, we select a range of core commercial metrics, including the click-through-rate per page view (pCTR), the gross merchandise volume (GMV), the averaged number of payments (ANP), the number of adding to cart (NAC), and the diversity metric.

Table 4: The performance improvements on Taobao App with online A/B test. The results are percentage numbers with "%" omitted.

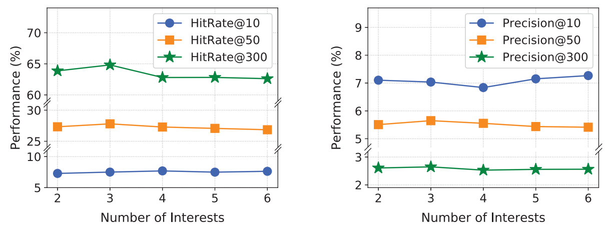
Method	pCTR	GMV	ANP	NAC	Diversity
Baseline	+0.0	+0.0	+0.0	+0.0	+0.0
UMI	**+0.95**	**+8.84**	**+5.35**	**+1.37**	**+1.79**

The A/B test results averaged over a week are reported in Table 4. Note that the results are reported with relative improvements. Here, it can be clearly observed that the proposed UMI provides better performance on all these commercial metrics. Especially, it outperforms the current system in terms of the GMV and ANP by a large margin. Another remarkable improvement is on the diversity metric, which is defined as the proportion of categories that are covered in the recommended items. Therefore, we can conclude that the proposed UMI generates better representations in characterizing precise and diverse interests of millions of users in Taobao.

4.4 Further Analysis

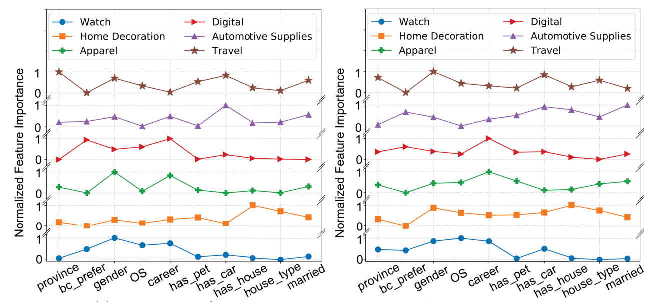
Number of Interests. We further investigate the model performance with respect to the parameter sensitivity. Specifically, we set the number of interests in the range of $[2, 7]$ in the MovieLens dataset, and report the hitrate@N and precision@N metrics where $N = 10, 50$, and 300. The results are illustrated in Figure 2. Here, we can observe that the overall performances are comparable when the number of interests varies. Generally, the model gets better performance when $K = 3$ in terms of HitRate@N, and when $K = 5/6$ in terms of Precision@N.

Noise Immunity in User Profile. In practical applications, some null or wrong values might be encountered in the user profiles, which generally result in performance degradation. However, in



(a) HitRate. (b) Precision.

Figure 2: Performance patterns by varying K values.



(a) GBDT weights. (b) UMI weights.

Figure 3: The weights of different user features generated by GBDT and UMI on different product categories.

Table 5: Comparison results of MIND and UMI against noise information in user profiles in Alibaba dataset. ▼ indicates the relative performance degradation. The results are percentage numbers with "%" omitted.

	Metrics@10		Metrics@50		Metrics@300	
	HitRate	Precision	HitRate	Precision	HitRate	Precision
MIND	4.262	1.526	11.203	0.802	26.799	0.320
MIND + Noise Data	3.886	1.392	10.581	0.714	24.676	0.287
▼%	-8.82	-8.78	-5.55	-10.97	-7.92	-10.31
UMI	5.454	1.953	13.605	0.974	30.628	0.366
UMI + Noise Data	5.239	1.842	13.364	0.941	30.019	0.345
▼%	**-3.94**	**-5.68**	**-1.77**	**-3.39**	**-1.99**	**-5.74**

comparison with existing methods, for example, MIND, which directly concatenates the raw user profile information, the proposed UMI can dynamically select the important features and filter out noise for different users through the IR module, making it more robust to user feature noise in comparison with other methods. To empirically verify its robustness, we randomly corrupt 10% features with noise in the more complex and difficult Alibaba dataset, and compare the performance degradation with MIND. The results are shown in Table 5. From the results, it can be observed that after being injected with noise data, both methods show certain performance decrements in both HitRate and Precision. However, in comparison with MIND which has fallen by about 10% in performance, the UMI shows significantly lower performance degradation, illustrating its better noise immunity.

Impact of User Profile. To illustrate the effect of user profile features in UMI, we investigate how UMI activates user features in different categories in Alibaba dataset. Firstly, we learn user feature importance in *different categories* through the GBDT model [6]. Specifically, for each product category, we train a GBDT model which takes user and item features as input, and predicts whether the user likes the item or not. The importance of different features are then extracted and visualized in Figure 3(a). As there are more than 80 primary categories and more than 30 user features in the Alibaba dataset, we choose 10 user profile features and 6 primary categories for clarity. Then, we plot the corresponding weights in Equation 7 generated by UMI in Figure 3(b). It is noted that different from the multiple GBDT models trained on different categories, our UMI is trained based on the samples collected from all the categories. For each feature, the mean value of the feature weights via feeding clicked samples of the same category is presented as the category-wise feature importance. Here, we can treat a primary category as the proxy for a user interest.

There are several observations from the results. First, different product categories naturally activate different user profile features. It can be seen that for both GBDTs and our proposed UMI, various features show different importance in different product categories. This is even an intuitive property in recommender systems, while may be overlooked by most multi-interest learning approaches. For example, the baseline models like MIND have static and identical user feature weights on different categories. Second, UMI dynamically identifies important features for different product categories. For example, in Figure 3(b), the feature "has_house" is highly activated in the home decoration category, while the feature "has_car" is strongly correlated with the user interests of "automotive supplies" and "travel". Note that the weights learned by UMI and the multiple GBDTs have a similar trend, indicating that UMI is effective to exploit user profile features to refine the user's multiple interests.

Case Study. In Figure 4, we randomly select two users from Taobao App, and list their recent clicked item sequences (shown in the left part), as well as the most relevant items retrieved by each interest (shown in the right part) from our UMI. Here, we pick four interests and take the dominating category of these items to express the semantic of the corresponding interest. For example, the first interest in Figure 4(a) is named as "Handbags" since the most relevant items retrieved by this interest fall in this category. It is clear to see that the four primary interests "Handbags", "Hats", "Jewelry" and "Dresses" for the user in Figure 4(a) and the four primary interests "Green Plants", "Automotive Supplies", "Mouse and Keyboards", and "Fish Tanks" for the user in Figure 4(b) are successfully discovered. Besides, in the left part, we also include the user-guided attention weight calculated by Equation 2 for each historical item (*i.e.,* the orange bar). It is obvious that the items with high weights are relevant to the user's primary interests, while the others like items (3), (9) and (10) for the user in Figure 4(a), and items (3), (6), and (9) for the user in Figure 4(b) are a bit messy.

Following the investigation made in Figure 3, we further pick a user and display the activation weights provided by the interest refinement module for the user profile features. Specifically, Figure 5 illustrates the corresponding relevance weights of several

Figure 4: Case study on two users in Taobao. The left part lists the sequence of the user click behaviors. The orange bar associated with each item shows the activation weight on the item. In the right part, the items retrieved by the user's four interests are displayed respectively. The interest semantics (*e.g.,* handbags) are determined according to the dominating category of the retrieved items (*Best viewed in color*).

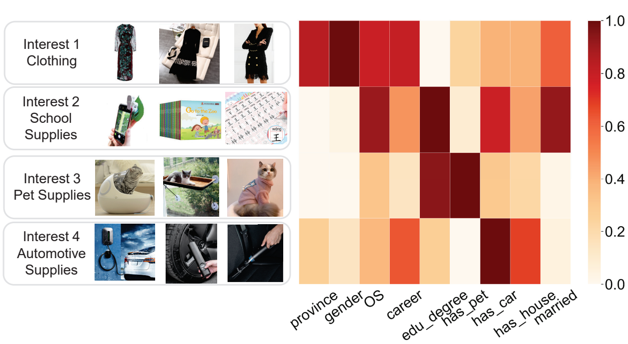


Figure 5: Visualization of the relevance weights derived by the interest refinement module (*Best viewed in color*).

user profile features including "user province", "gender", "mobile OS", "career", "education degree", "has pet/car/house or not", and "married", as well as the historically clicked items relevant to each interest. From Figure 5, we can see that different interests focus on different profile features, resulting in more precise and diverse interest learning. For example, the user clicked items of automotive supplies, and this interest is highly related to the feature "has_car". It is worthwhile to mention that several home charging station items are also included in the user's historical behaviors. Hence, the interest in "Automotive Supplies" also highly activates the feature "has_house" for the user. Besides, interests in clothing, school supplies, and pet supplies also generate high relevance weights on features "gender", "edu_degree", and "has_pet", showing superior ability in identifying different user profile features for different interests, leading to better user multi-interest understanding.

5 CONCLUSION

In this paper, a novel user-aware multi-interest learning framework is proposed for user interest understanding, enabling more precise and diverse candidate matching in recommenders. It emphasizes that 1) user's multiple preferences are not only a simple reflection of their behaviors, but should be inherently regulated by the user profile information, and 2) a static user profile representation is insufficient for multi-interest learning. Thus, different historical behaviors and user profile features should be jointly exploited to reflect user intentions. Based on this idea, we introduce a dual-attention routing and an interest-refinement mechanism to learn multiple user interests. Besides, an HN strategy is devised, which can significantly enhance the discriminative learning for the model and be generally applicable to the current multi-interest modeling frameworks. The offline experiments through two real-world datasets confirm the superiority of the proposed method. Moreover, the online results demonstrate that our UMI achieves improvements in terms of a range of commercial metrics. UMI now has been deployed in Taobao App for large-scale item recommendation and serves hundreds of millions of users.

ACKNOWLEDGMENTS

This work was supported by Alibaba Group through Alibaba Research Intern Program.

REFERENCES

[1] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2942–2951.

[2] Qiwei Chen, Huan Zhao, Wei Li, Pipei Huang, and Wenwu Ou. 2019. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*. 1–4.

[3] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *ACM Conference on Recommender Systems*. 191–198.

[4] Marzieh Edraki, Nazanin Rahnavard, and Mubarak Shah. 2020. Subspace capsule network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 10745–10753.

[5] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep session interest network for click-through rate prediction. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2301–2307.

[6] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[7] Carlos A Gomez-Uribe and Neil Hunt. 2015. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems* 6, 4 (2015), 1–19.

[8] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *International World Wide Web Conference*. 173–182.

[9] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. *International Conference on Learning Representations*.

[10] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2553–2561.

[11] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *ACM International Conference on Information and Knowledge Management*. 2333–2338.

[12] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2021), 535–547.

[13] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

[14] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *ACM International Conference on Information and Knowledge Management*. 2615–2623.

[15] Chenliang Li, Cong Quan, Li Peng, Yunwei Qi, Yuming Deng, and Libing Wu. 2019. A capsule network for recommendation and explaining what you like and dislike. In *ACM SIGIR Conference on Research and Development in Information Retrieval*. 275–284.

[16] Houyi Li, Zhihong Chen, Chenliang Li, Rong Xiao, Hongbo Deng, Peng Zhang, Yongchao Liu, and Haihong Tang. 2021. Path-based Deep Network for Candidate Item Matching in Recommenders. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1493–1502.

[17] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *International World Wide Web Conference*. 689–698.

[18] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *International Conference on Learning Representations*.

[19] Tianyi Liu, Xiangyu Lin, Weijia Jia, Mingliang Zhou, and Wei Zhao. 2020. Regularized Attentive Capsule Network for Overlapped Relation Extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*. 6388–6398.

[20] Yujie Lu, Shengyu Zhang, Yingxuan Huang, Luyao Wang, Xinyao Yu, Zhou Zhao, and Fei Wu. 2021. Future-Aware Diverse Trends Framework for Recommendation. In *International World Wide Web Conference*. 2992–3001.

[21] Fuyu Lv, Taiwei Jin, Changlong Yu, Fei Sun, Quan Lin, Keping Yang, and Wilfred Ng. 2019. SDM: Sequential deep matching model for online large-scale recommender system. In *ACM International Conference on Information and Knowledge Management*. 2635–2643.

[22] Ze Lyu, Yu Dong, Chengfu Huo, and Weijun Ren. 2020. Deep match to rank model for personalized click-through rate prediction. In *AAAI Conference on Artificial Intelligence*, Vol. 34. 156–163.

[23] Aditya Pal, Chantat Eksombatchai, Yitong Zhou, Bo Zhao, Charles Rosenberg, and Jure Leskovec. 2020. PinnerSage: multi-modal user embedding framework for recommendations at pinterest. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2311–2320.

[24] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic Routing Between Capsules. In *Advances in Neural Information Processing Systems*, Vol. 30.

[25] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *International World Wide Web Conference*. 285–295.

[26] Qiaoyu Tan, Jianwei Zhang, Jiangchao Yao, Ninghao Liu, Jingren Zhou, Hongxia Yang, and Xia Hu. 2021. Sparse-interest network for sequential recommendation. In *ACM International Conference on Web Search and Data Mining*. 598–606.

[27] Zhibo Xiao, Luwei Yang, Wen Jiang, Yi Wei, Yi Hu, and Hao Wang. 2020. Deep Multi-Interest Network for Click-through Rate Prediction. In *ACM International Conference on Information and Knowledge Management*. 2265–2268.

[28] Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, and Ed H Chi. 2020. Mixed negative sampling for learning two-tower neural networks in recommendations. In *International World Wide Web Conference*. 441–447.

[29] Xinsong Zhang, Pengshuai Li, Weijia Jia, and Hai Zhao. 2019. Multi-labeled relation extraction with attentive capsule network. In *AAAI Conference on Artificial Intelligence*, Vol. 33. 7484–7491.

[30] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *AAAI Conference on Artificial Intelligence*, Vol. 33. 5941–5948.

[31] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1059–1068.

[32] Han Zhu, Daqing Chang, Ziru Xu, Pengye Zhang, Xiang Li, Jie He, Han Li, Jian Xu, and Kun Gai. 2019. Joint optimization of tree-based index and deep model for recommender systems. In *Advances in Neural Information Processing Systems*. 3971–3980.

[33] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. 2018. Learning tree-based deep model for recommender systems. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1079–1088.