Multi-interest Diversification for End-to-end Sequential Recommendation

WANYU CHEN, National University of Defense Technology, China and University of Amsterdam, The Netherlands
PENGJIE REN, Shangdong University, China
FEI CAI, National University of Defense Technology, China
FEI SUN, Alibaba Group, China
MAARTEN DE RIJKE, University of Amsterdam, The Netherlands and Ahold Delhaize, The Netherlands

Sequential recommenders capture dynamic aspects of users' interests by modeling sequential behavior. Previous studies on sequential recommendations mostly aim to identify users' main recent interests to optimize the recommendation accuracy; they often neglect the fact that users display multiple interests over extended periods of time, which could be used to improve the diversity of lists of recommended items. Existing work related to diversified recommendation typically assumes that users' preferences are static and depend on post-processing the candidate list of recommended items. However, those conditions are not suitable when applied to sequential recommendations. We tackle sequential recommendation as a list generation process and propose a unified approach to take accuracy as well as diversity into consideration, called *multi-interest,*

20

A preliminary version of this article appeared in the proceedings of CIKM 2020 [10]. In this extension, we (1) propose another interest extractor, i.e., dynamic routing, in the implicit interest mining module, and another type of disagreement regularization, i.e., output disagreement regularization, in our interest-aware, diversity promoting loss; (2) investigate the performance of our multi-interest, diversified, sequential recommendation model with different interest extractors in implicit interest mining, i.e., multi-head attention vs. dynamic routing; (3) investigate the performance of multi-interest, diversified, sequential recommendation with various latent interests numbers; (4) explore the influence of the parameter λ on the performance of multi-interest, diversified, sequential recommendation; (5) investigate the performance of multi-interest, diversified, sequential recommendation with different types of disagreement regularization; (6) investigate the impact of maximum entropy regularization on the performance of multi-interest, diversified, sequential recommendation; (7) provide a case study to show the recommendations generated by multi-interest, diversified, sequential recommendation; (8) analyze the computational complexity of the baseline model as well as our proposal; and (9) survey more related work and conduct a more detailed analysis of the approach and experimental results.

This research was (partially) funded by the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl.

All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

Authors' addresses: W. Chen, Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, 410073, China, University of Amsterdam, Amsterdam, The Netherlands; email: wanyuchen@nudt.edu.cn; P. Ren, Shangdong University, Qingdao, 266000, China; email: jay.ren@outlook.com; F. Cai (corresponding author), Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, 410073, Changsha, China; email: caifei@nudt.edu.cn; F. Sun, Alibaba Group, Beijing, 100000, China; email: ofey.sf@alibaba-inc.com; M. de Rijke, University of Amsterdam, Amsterdam, The Netherlands, Ahold Delhaize, Zaandam, The Netherlands; email: m.derijke@uva.nl.

1046-8188/2021/08-ART20 $15.00
https://doi.org/10.1145/3475768

diversified, sequential recommendation. Particularly, an implicit interest mining module is first used to mine users' multiple interests, which are reflected in users' sequential behavior. Then an interest-aware, diversity promoting decoder is designed to produce recommendations that cover those interests. For training, we introduce an interest-aware, diversity promoting loss function that can supervise the model to learn to recommend accurate as well as diversified items. We conduct comprehensive experiments on four public datasets and the results show that our proposal outperforms state-of-the-art methods regarding diversity while producing comparable or better accuracy for sequential recommendation.

CCS Concepts: • **Recommender systems**;

Additional Key Words and Phrases: Sequential recommendation, diversified recommendation

ACM Reference format:
Wanyu Chen, Pengjie Ren, Fei Cai, Fei Sun, and Maarten de Rijke. 2021. Multi-interest Diversification for End-to-end Sequential Recommendation. *ACM Trans. Inf. Syst.* 40, 1, Article 20 (August 2021), 30 pages.
https://doi.org/10.1145/3475768

1 INTRODUCTION

Methods for conventional recommendation, e.g., collaborative filtering–based methods [44], often mix all of a user's historical behaviors and suppose that user preferences are static. Such methods ignore the sequential signals underlying user behavior and thus cannot capture users' dynamic preferences [35]. Sequential recommendation has been proposed to tackle those characteristics; it is aimed at predicting a user's next interaction based on modeling his previous sequential interactions [39].

Previous research into **sequential recommendation (SR)** is typically based on neural models, e.g., **recurrent neural network (RNN)** [16], **convolutional neural network (CNN)** [47], or transformer [24] based. Recently, many factors have been taken into consideration to help improve the performance of sequential recommendation, e.g., personalization [40], repeat consumption [42], context [41], and collaboration [49]. Most of those factors are assumed to help improve the accuracy of recommendations only, while ignoring the diversity of the recommendation list. However, research has shown that diversity is an important factor that may influence the performance of a recommender system, since users may prefer more diverse recommendations [58].

Users may have multiple interests in terms of different categories or themes of items, especially when we consider behavior sequences that span long periods of time. For example, Figure 1 shows the sequential watching behavior of a user who has expressed most interest in "Animation," "Action," and "Adventure" (on the left-hand side); the most recent watched movie also belongs to those categories. According to previous approaches that target at recommendation accuracy, the list of recommendations should be full of movies in the three genres, e.g., the list of recommendations shown in the center in Figure 1. However, this strategy ignores the fact that the user has also watched "Family," "Drama," and "Sci-Fi" movies occasionally. We hypothesize that it will be more effective to provide the user with recommendations that can potentially satisfy all these interests rather than to focus on one particular interest only. Concretely, in Figure 1, we recommend a list including movies covering all those categories simultaneously as in the list on the far right-hand side. Additionally, users are often exploratory and they may not have a specific goal in mind. If users are always provided with homogeneous lists of recommendations, then they may get bored and are not satisfied [45].

There has been much research into diversification in conventional recommendation tasks [52] and web search scenarios [1, 31, 36]. These approaches typically aim to improve the diversity of recommendations by reranking the items in a candidate list of recommended items produced by

Fig. 1. An example of a user's past movie watching behavior (left) together with two kinds of sequential recommendation: without diversification (right) and with diversification (far right).

general recommendation models. However, those approaches are not suitable for sequential recommendations (SRs). On the one hand, they regard user interest as static and fixed beforehand, which is unworkable in most SR scenarios [9]. On the other hand, these reranking approaches achieve accuracy and diversification of a list of recommended items in separate steps. They first generate a candidate list of recommended items with a general recommendation model. Then, items in the candidate list are reranked based on their relevance scores as well as diversity scores [27, 52]. Thus, the reranking performance relies heavily on the candidate list of recommended items. However, since the general recommendation models are merely focused on accuracy during training, it is difficult to design an optimal reranking strategy for different approaches. Besides, these general recommendation models typically represent each user by a fixed-length vector [28]; such vectors are hard to use for modeling users' multiple interests reflected in their historical behavior, as the dimensionality of the vector is much smaller than the number of items in whole dataset.

In this article, we consider *accuracy* and *diversity* for SR simultaneously. We formulate SR as a *sequence to list process* and model a user's previous sequential behavior and the relationship among recommended items in a unified framework. To this end, we propose an end-to-end **multi-interest, diversified, sequential recommendation (MDSR)** model. An **implicit interest mining (IIM)** module is first introduced to capture a user's multiple latent interests from their historical sequential behavior and then an **interest-aware, diversity promotin (IDP)** decoder is applied to produce a list of recommended items to cover those interests. We also design an IDP loss function to supervise the learning of the IIM module and an IDP decoder so as to let the model take recommendation accuracy as well as diversity into consideration while training.

More specifically, we first encode a user's past sequential behavior into latent representations with a general sequence encoder, i.e., a **gated recurrent unit (GRU)**. Then, the IIM module extracts users' multiple interests. We study two strategies for the IIM module: (1) a dynamic routing algorithm from a capsule network where each high-level capsule represents a latent interest and (2) a multi-head attention mechanism where each head models a particular latent user interest. Finally, we design an IDP decoder to generate a list of recommended items. It selects recommended items one by one to gradually satisfy the multiple interests and thus promote diversity of the whole list of recommended items. When deciding which item, to recommend next, the IDP decoder takes previous recommended items as input and evaluates to what extent each interest is satisfied by the items that have already been recommended. To train the entire model, we propose an IDP loss function that consists of four parts: (1) a cross-entropy loss to help the model learn to recommend accurate items; (2) a diversity loss based on a self-critic strategy; since we do not have a ground-truth list as a supervision signal for diversity, we design a self-critic strategy to help the IDP decoder learn to generate diversified recommendations; specifically, we reward our model when the generated recommendation list is more diverse than the list produced by the conventional strategy (the rank-by-score strategy); otherwise, we punish the model; (3) disagreement regularization to distinguish different interests; and (4) maximum entropy regularization to avoid cases where one interest dominates.

To assess the performance of MDSR, we perform comprehensive experiments on four public datasets, i.e., two movie recommendation dataset and two e-commerce datasets. The experimental results show that multi-interest, diversified, sequential recommendation (MDSR) outperforms state-of-the-art baselines in terms of both accuracy as well as diversity metrics.

To summarize, in this article:

- We tackle the problem of sequential recommendation by simultaneously considering recommendation accuracy as well as diversity, and we propose an MDSR method that is the first end-to-end list generation-based neural framework for SR.
- We design an implicit interest mining module with two different interest extractors, i.e., multi-head attention and dynamic routing, to extract latent multiple user interests from user behavior.
- We design an interest-aware, diversity promoting decoder to generate diversified recommendations and an interest-aware, diversity promoting loss function to supervise the learning of our model.
- We conduct extensive experiments on movie recommendation datasets and e-commerce recommendation datasets to prove the effectiveness of MDSR and analyze the impact of each component of MDSR.

2 RELATED WORK

In this section, we survey related work on (1) sequential recommendation, (2) diversified recommendation, (3) capsule networks, and (4) attention mechanisms.

2.1 Sequential Recommendation

Early approaches for SR are typically based on Markov chains [62], which are not suitable when dealing with long sequences [7]. Recently, neural models have been shown to be effective for the sequential recommendation task. Hidasi et al. [17] first introduce recurrent neural networks (RNNs) into SR and propose a session-parallel training mechanism. Following this work, variant models based on RNNs have been proposed for SR. Hidasi et al. [18] propose a parallel RNN structure to model sessions with clicks and features of the interacted items. Quadrana et al. [40] introduce a hierarchical RNN model to capture users' cross-session information. Besides these RNN-based proposals, there is work based on other neural network structures. Xu et al. [54] capture long-term and short-term dependencies among user behavior with a recurrent convolutional neural network. Memory networks have also been applied in SR. Chen et al. [11] store users' historical behavior in a user memory network and apply an attention mechanism to capture a user's current preference for SR. The transformer model has also been applied to the sequential recommendation task. Kang and McAuley [24] use a two-layer transformer structure [48] to model a user's previous sequential behavior. Tang and Wang [47] propose an approach for top-N sequential recommendation by modeling recent actions as an "image" among time.

The studies mentioned so far all aim to improve the accuracy of a list of recommended items. They typically use an overall representation of each user, i.e., a vector with a fixed length, which is not enough to express user multiple interests [28]. Recently, Wang et al. [51] have proposed **mixed-channel purpose routing networks (MCPRNs)**, which model the interacted items in a sequence with a mixture-channel RNN so as to capture users' multiple interests; all channel representations are integrated together to recommend the next item. For supervising the training process, this model only uses the cross-entropy loss that is aimed at improving the recommendation accuracy. There are no signals that help the model to learn to distinguish these interests and generate a diversified recommendation list.

The aforementioned approaches all apply the *rank-by-score* strategy and output the recommendation list by ranking items according to their scores, which is different from our list-generation process and not able to capture the relationship among recommended items in the list. Our own recent work [10] also models users' multiple intents behind their sequential behavior. However, in this article we propose another interest extractor, i.e., dynamic routing, in the IIM module, to capture a user's multiple interests. We also investigate the performance of our multi-interest, diversified, sequential recommendation model with different interest extractors in implicit interest mining (IIM), i.e., multi-head attention vs. dynamic routing, and find that MDSR shows better performance in terms of accuracy with multi-head attention than with dynamic routing in IIM module.

In this article, we address the recommendation accuracy and diversification for SR simultaneously and propose an end-to-end framework. An IIM module is first applied to explore a user's multiple interests and an IDP decoder is then used to produce diversified recommendations to satisfy those interests. We also design an interest-aware, diversity promoting (IDP) loss, that included accuracy and diversity losses, to supervise the learning process of the model that we propose.

2.2 Diversified Recommendation

It has been found that striving for recommendation accuracy might result in homogeneous recommendations, since the items with high accuracy tend to have similar content and/or genres [37]. Recommendation diversification and search results diversification have long been a vital research topic. There are generally two solution directions for recommendation diversity: *aggregated diversity* and *individual diversity*. The former is meant to increase the exposure of long-tail items so as to achieve global recommendation diversity for all items [6, 25, 37], while the later aims to improve the diversity of items recommended to an individual user. In this work, we focus on the latter one.

The most widely used approach to diversification is **maximal marginal relevance (MMR)** [8], which is a reranking algorithm. First, maximal marginal relevance (MMR) builds a similarity matrix between each pair of candidate items and then it iteratively selects K items with maximum marginal relevance to form the final list of recommended items. The maximum marginal relevance is composed of relevance and diversity scores. The relevance score can be determined by a general recommendation model while the diversity score can be calculated based on the similarity matrix. In this way, MMR can return a diversified list of recommended items. Qin and Zhu [38] design an objective function that contains an entropy regularizer to improve the diversity of recommendations. They prove the monotonicity and submodularity of the objective function and use a greedy algorithm to optimize it. Sha et al. [45] also design an objective function that takes relevance as well as diversity of candidate items into consideration. Some **learning to rank (LTR)** approaches have also been proposed and applied to the task of diversifying recommendations. Cheng et al. [12] introduce a diversified collaborative filtering model to learn to generate accurate as well as diversified recommendations. However, those learning to rank (LTR)-based approaches need a ground-truth diversified recommendation list for learning, which is unavailable in most recommendation scenarios. Hu et al. [21] propose a personalized session-based recommendation model with shallow wide-in-wide-out networks. They take the user embedding and the user-related sessions as inputs to generate recommendations. This work aims to provide different user-session context with different recommendations and the diversity evaluation in this work is to measure the mean non-overlap ratio between each pair of recommendation lists. Thus it is different from our individual diversification task. Kim et al. [25] propose a sequential and diverse recommendation model that predicts a ranked list containing general as well as tail items, i.e., S-DIV.

Current approaches typically achieve recommendation diversification based on reranking a candidate item, set produced by a general recommendation model. Instead, our proposed MDSR can integrate the sequential behavior modeling and diversified recommendations generation in an end-to-end model. Our experimental results below demonstrate that MDSR performs competitively. Wang et al. [50] provide a review on the challenges and recent progress of sequential recommender systems. They also point that leveraging knowledge from different domains can help to generate more diverse recommendations, which can be a direction for our future work.

A preliminary version of our work [10] also aims to improve recommendation diversity in an end-to-end approach to SR, i.e., an **intent-aware, diversified, sequential recommendation (IDSR)** model. We improve upon our prior work by incorporating a new interest extractor in the implicit interest mining module, i.e., dynamic routing, and a new type of disagreement regularization in our interest-aware, diversity promoting loss, i.e., output disagreement regularization.

2.3 Capsule Networks

A capsule is a group of neurons whose activity vector indicates the instantiation parameters of a certain type of entity [19]. The length of the activity vector is used to represent the probability that the entity exists. Often two-levels of capsules are used, where the active capsules at the low-level make predictions for the instantiation parameters of the high-level capsules. When most predictions agree, a high-level capsule can be activated. Instead of using backpropagation, dynamic routing has been proposed to learn the weights by connecting low-level and high-level capsules in an iterative way using the Expectation-Maximization algorithm [20]. Sabour et al. [43] show that capsule networks can help to learn representations containing richer information than convolutional neural networks (CNNs) in computer vision. Capsule networks with dynamic routing also show their effectiveness in capturing multiple labels or aspects in text sequences. Yang et al. [55] investigate the effectiveness of capsule networks applied to text classification tasks. Xia et al. [53] explore capsule-based architectures to extract semantic features from utterances and aggregate them to discriminate diversely expressed intents. Zhao et al. [60] propose a framework to optimize the routing processes and show competitive performance in multi-label text classification tasks. Li et al. [28] introduce capsules and dynamic routing into a recommender system to capture users' multiple interests in e-commerce recommendations.

In this work, we apply capsule networks with dynamic routing to extract representations of a user's multiple interests in an IIM module. Our work differs from the research described above, since the prior work aggregates interests for making accurate recommendations, while we use interest representations to generate diverse as well as accurate recommendations.

2.4 Attention Mechanisms

Attention mechanisms have been widely used in recommender systems to help capture users' main interests based on which to improve recommendation accuracy. Li et al. [29] propose a **neural attentive session-based recommendation machine (NARM)** that employs the last hidden state from the session-based RNN to attend to the previous interactions to identify users' main interests. Recently, self-attention mechanisms and transformer-based architectures have also been

Table 1. Summary of the Main Notation Used in the Article

Notation	Description
$\mathbf{x}_t$	the embedding of item, x_t
F_u	user's global representation
F_u^i	user's ith interest representation
$\boldsymbol{h}_t$	hidden representation at timestep t in GRU of sequence encoder
$\boldsymbol{h}_{t-1}^y$	hidden representation at timestep $t-1$ in GRU of IDP decoder
$S(v_i)$	the final score of item, v_i generated by Equation (2)
$\boldsymbol{M}(R_L)$	the diversity score of recommendation list R_L in IDP loss
$\boldsymbol{A}^i$	the attention distribution produced by interest i in Equation (6)
$\boldsymbol{W_z}$,$\boldsymbol{W_r}$,$\boldsymbol{W_h}$	parameters in GRU of sequence encoder
$\boldsymbol{W}_i^Q, \boldsymbol{W}_i^K, \boldsymbol{W}_i^V$	parameters in multi-head attention-based IIM
M_{ij}	bilinear mapping matrix in dynamic routing-based IIM
b_{ij}	the routing logit in dynamic routing-based IIM
$\boldsymbol{B}$	bilinear mapping matrix in IDP decoder
$\boldsymbol{W}^w$	transform matrix in IDP decoder
$\boldsymbol{W_y}$,$\boldsymbol{W_A}$,$\boldsymbol{W_B}$	parameters in the attention mechanism of IDP decoder
λ	hyper-parameter controlling the balance between relevance score and diversity score in Equation (2)
λ_e	hyper-parameter controlling the contribution of accuracy loss in Equation (16)

applied to recommender systems [48]. Kang and McAuley [24] apply a multi-head self-attention mechanism to capture users' main interests based on their sequential behavior, and Sun et al. [46] introduce a bidirectional representation learning method based on the transformers for SR and achieve satisfied performance for sequential recommendation. They all help to improve recommendation performance in terms of accuracy.

However, the studies listed above apply attention mechanisms to capture users' *main* interests instead of *multiple* interests. Both our work in this article and the intent-aware, diversified, sequential recommendation (IDSR) model proposed in Reference [10] apply a multi-head attention mechanism to help extract multiple interests behind users' sequential behavior with each attention distribution indicating a particular user interest, which is then used to help generate diversified recommendations covering those interests.

3 MULTI-INTEREST, DIVERSIFIED SEQUENTIAL RECOMMENDATIONS

3.1 Overview

In this section, we introduce our proposed model, MDSR, in detail. Table 1 summarizes the main notation used in the article. Let a user u and their sequential behavior denoted as $X_u = \{x_1, x_2, \ldots, x_T\}$ be given, where each x_i indicates an item, that u interacted with, such as an add-to-cart or purchase action in an e-commerce scenario. We aim to provide the user with a list of recommended items to predict the item, that they may interact with next. We hope to capture the user's multiple interests reflected in X_u and thus the list of recommended items should be accurate as well as diverse so as to cover those interests.

Most existing approaches for SR focus on modeling a user's main interest and represent the user with a fixed-length vector. In this article, it is supposed that there are M latent interests reflected by each user behavior sequence, i.e., $A = \{a_1, \dots, a_M\}$. We hope to recommend a list of items to satisfy all of those interests, which can be formulated as

$$P(R_L \mid u, X_u) = \sum_{m=1}^{M} P(a_m \mid u) P(R_L \mid a_m, u, X_u), \tag{1}$$

where $P(a_m \mid u)$ indicates the importance of interest a_m to user u; $P(R_L \mid a_m, u, X_u)$ denotes the probability that evaluates to what extent the interest a_m is satisfied by the current recommendation list R_L.

Theoretically, finding an exact solution to maximizing $P(R_L \mid u, X_u)$ is an NP-hard problem because of the huge search space [3, 8]. However, we can approximate it by designing a greedy selection algorithm. The idea, is that we can first initialize R_L as an empty set and then in each step we select one item, and add it to R_L, where the selected item, at step t can be obtained by a scoring function:

$$\begin{aligned} v_t &= \arg\max_{v \in V \setminus R_{t-1}} S(v), \\ S(v) &= \lambda P(v \mid u, X_u) + (1-\lambda) \sum_{m=1}^{M} P(v \mid a_m) W(\overline{R_{t-1}}, a_m), \end{aligned} \tag{2}$$

where V denotes the set of all items; R_{t-1} is the list of recommended items produced by the previous $t-1$ steps; $v \in V \setminus R_{t-1}$ ensures that the final list of recommended items contains no repeated items. The score $S(v)$ contains two parts, which are controlled by a hyper-parameter λ; $P(v \mid u, X_u)$ denotes the relevance score of item, v to u based on the current behavior sequence; $P(v \mid a_m)$ indicates the relevance of item, v to interest a_m; $W(\overline{R_{t-1}}, a_m)$ evaluates to what extent the recommended items in R_{t-1} cannot satisfy interest a_m. Before calculating the relevance score (left part of Equation (2)) and the diversity score (right part of Equation (2)), we do normalization across all candidate items.

Based on Equation (2), we propose an end-to-end neural framework, i.e., MDSR, to model the user's sequential behavior and generate diversified recommendations simultaneously. We show a graphical representation of the main framework in Figure 2. There are three major components in MDSR, i.e., a *sequence encoder*, an IIM *module*, and an IDP *decoder*. More specifically, we first apply a sequence encoder to those sequential interactions and generate a hidden state of each timestep. Then we propose an IIM module to extract users' multiple interests and output latent representations of those interests, which are shown with different colors in Figure 2. The IDP decoder is finally used to produce recommendations according to Equation (2) by selecting items to satisfy and cover those interests gradually. For training, we design an IDP loss to supervise the model to learn to generate accurate as well as diversified recommendations. In the following sections, we give detailed introductions of each component.

3.2 Sequence Encoder

Although the sequence encoder is the base of the whole framework, it is not the focus of our model. We choose the most commonly used one as our sequence encoder, i.e., a gated recurrent unit (GRU) [17]. It should be noted that other sequence encoders can also be adopted in our model.

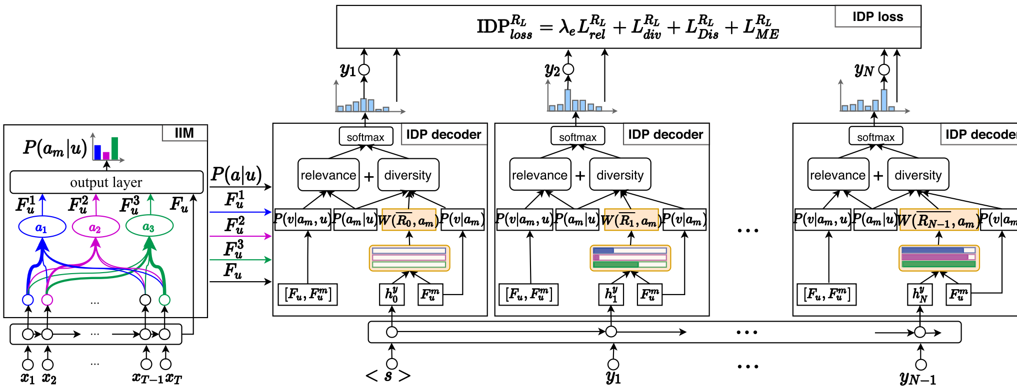


Fig. 2. Framework of MDSR. Blue, purple, and green represent different user interests.

The sequence encoder in this article can be formulated as

$$\begin{aligned} z_t &= \sigma\left(W_z[x_t, h_{t-1}]\right) \\ r_t &= \sigma\left(W_r[x_t, h_{t-1}]\right) \\ \hat{h}_t &= \tanh(W_h[x_t, r_t \odot h_{t-1}]) \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t, \end{aligned} \tag{3}$$

where $\boldsymbol{x}_t$ is the embedding of x_t; $\boldsymbol{W}_z$, $\boldsymbol{W}_r$ and $\boldsymbol{W}_h$ denote the weight parameters in GRU. These hidden representations, i.e., $\{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_T\}$, are then stacked into a matrix $\boldsymbol{H}_S \in \mathbb{R}^{T \times d_e}$. Following [29], the last behavior often plays an important role in predicting a user's next interaction. Thus, we regard the last hidden representation $\boldsymbol{h}_T$ as the user's global representation:

$$F_u = \boldsymbol{h}_T. \tag{4}$$

3.3 IIM Module

The IIM module aims to extract users' multiple interests as reflected in their behavior sequences. Intuitively, different interactions in a user's sequential behavior can express their different interests. Some interactions are more representative for a certain interest than others. For example, the second and last two actions in Figure 1 are more obvious in reflecting the user's interest in "Animation," "Action," and "Adventure" than other interactions. Motivated by this observation, we explore two methods to extract a user's multiple interests, i.e., multi-head attention and dynamic routing.

3.3.1 Multi-Head Attention-based IIM. We apply a multi-interest attention mechanism where each attention function explores one certain interest. First, we project $\boldsymbol{H}_S$ and F_u into M spaces. M attention functions are then employed in parallel to calculate the user's M interest representations $\{F^1_u, F^2_u, \ldots, F^M_u\}$:

$$F^i_u = \mathrm{Attention}\left(F_u \boldsymbol{W}^Q_i, \boldsymbol{H}_S \boldsymbol{W}^K_i, \boldsymbol{H}_S \boldsymbol{W}^V_i\right), \tag{5}$$

where $\boldsymbol{W}^Q_i \in \mathbb{R}^{d_e \times d}$, $\boldsymbol{W}^K_i \in \mathbb{R}^{d_e \times d}$, and $\boldsymbol{W}^V_i \in \mathbb{R}^{d_e \times d}$ are learnable parameters. The scaled dot-product attention [48] is applied as

$$\mathrm{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \boldsymbol{A}\boldsymbol{V} = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d}}\right)\boldsymbol{V}, \tag{6}$$

where $\boldsymbol{A}$ is the attention distribution of each interest.

3.3.2 Dynamic Routing-based IIM. The dynamic routing method, which was first introduced in CapsNet [43], is widely used for representation learning of capsules. It learns the weights on the connections between capsules using the Expectation-Maximization algorithm. In this work, we regard the item, embeddings of user sequences as low-level capsules and multiple interests as high-level capsules. The goal is to learn the high-level representations as well as low-level capsules iteratively. In each iteration, the routing logit b_{ij}, which represents the log prior probability that capsule i is coupled to capsule j, can be calculated by the inner product of the corresponding vectors of capsule i and j, i.e., $\boldsymbol{x}_i$ and F_u^j as follows:

$$b_{ij} = (F_u^j)^\top M_{ij}\boldsymbol{x}_i, \tag{7}$$

where M_{ij} is a bilinear mapping matrix. We can then calculate the total input to the high-level capsule j as a weighted sum of all low-level capsules:

$$z_j = \sum_{i=1}^{T} c_{ij} M_{ij}\boldsymbol{x}_i, \tag{8}$$

where c_{ij} denotes the weight for linking low-level capsule i and high-level capsule j. The weights between capsule i and all high-level capsules should add up to 1. Thus, we use a softmax with routing logit b_{ij} to calculate c_{ij}:

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_{m=1}^{M}\exp(b_{im})}. \tag{9}$$

As we expect the length of the output vector of a capsule to indicate the probability of the corresponding entity being present in the current input, a non-linear "squashing" function [43] is proposed to ensure that short vectors get shrunk to nearly zero length and long vectors get shrunk to a length close to 1. Thus, the vector of high-level capsule j is calculated based on z_j as follows:

$$F_u^j = \mathrm{squash}(z_j) = \frac{\left\|z_j\right\|^2}{1+\left\|z_j\right\|^2}\cdot\frac{z_j}{\left\|z_j\right\|}. \tag{10}$$

To calculate the high-level capsule representation F_u^j, we need to calculate a probability distribution with the inner production of F_u^j and $\boldsymbol{x}_i$ (see Equation (7)), which means the calculation of F_u^j relies on itself. Thus, dynamic routing is proposed to solve this problem in an iterative way. The dynamic routing algorithm is listed in Algorithm 1. The values of b_{ij} are initialized as zeros and the routing process is typically repeated three times to converge. After routing, we can get the representations of high-level capsules, i.e., multiple interest representations $\{F_u^1, F_u^2, \ldots, F_u^M\}$.

To stabilize and accelerate training, we apply layer normalization [5] on the inputs of multi-head attention as well as dynamic routing. Unlike batch normalization [22], the data used in layer normalization are independent of other samples in the same batch. Supposing the input is a vector $\mathbf{x}$ that contains all features of a sample, the operation is defined as

$$\mathrm{LayerNorm}(\mathbf{x}) = \alpha \odot \frac{\mathbf{x}-\mu}{\sqrt{\sigma^2+\epsilon}} + \beta, \tag{11}$$

where $\odot$ indicates an element-wise product, μ and σ are the mean and variance of $\mathbf{x}$, α, and β denote learned scaling factors and bias term.

ALGORITHM 1: Dynamic routing.

Input: R: iteration times;
$\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$: item, embeddings in a user sequence;
Output: $\{F_u^1, F_u^2, \ldots, F_u^M\}$: multiple interest representations;

```
for each low-level capsule i and high-level capsule j, initialize b_ij = 0;
for iter, in range (R) do
    for each low-level capsule i: c_ij = exp(b_ij) / sum_{m=1}^{M} exp(b_im).
    for each high-level capsule j: z_j = sum_{i=1}^{T} c_ij M_ij x_i.
    for each high-level capsule j: F_u^j = squash(z_j).
    for each low-level capsule i and each high-level capsule j: b_ij = b_ij + (F_u^j)^T M_ij x_i.
end for
return {F_u^1, F_u^2, ..., F_u^M}.
```

for each low-level capsule i and high-level capsule j, initialize $b_{ij} = 0$;
for iter, in range (R) **do**
for each low-level capsule i: $c_{ij} = \frac{\exp(b_{ij})}{\sum_{m=1}^{M} \exp(b_{im})}$.
for each high-level capsule j: $z_j = \sum_{i=1}^{T} c_{ij} M_{ij} \boldsymbol{x}_i$.
for each high-level capsule j: $F_u^j = \text{squash}(z_j)$.
for each low-level capsule i and each high-level capsule j: $b_{ij} = b_{ij} + (F_u^j)^\top M_{ij} \boldsymbol{x}_i$.
end for
return $\{F_u^1, F_u^2, \ldots, F_u^M\}$.

3.4 IDP Decoder

The IDP decoder is designed to generate the list of recommended items with the latent interests extracted by the IIM module of the previous section. According to Equation (2), we first calculate the relevance score of item, x as follows:

$$\begin{aligned} P(x_n \mid u, X_u) &= \frac{\boldsymbol{S}_{\boldsymbol{x}_n}}{\sum_{j=1}^{|V|} \boldsymbol{S}_{\boldsymbol{x}_j}} \\ S_{x_n} &= \sum_{m=1}^{M} S_{x_n}^m \\ S_{x_n}^m &= P(a_m \mid u) P(x_n \mid a_m, u) \\ P(x_n \mid a_m, u) &= \text{softmax}(\boldsymbol{x}_{\boldsymbol{n}}^\top \boldsymbol{B}[F_u, F_u^m]), \end{aligned} \tag{12}$$

where $\boldsymbol{B}$ denotes a bilinear parameter; $\boldsymbol{x}_{\boldsymbol{n}}$ is the embedding of the item, x_n; and $S_{x_n}^m$ denotes the relevance score of item, x_n to interest a_m. When calculating $S_{x_n}^m$, we also consider the importance of a particular interest a_m, i.e., $P(a_m \mid u)$, which can be generated by

$$P(a_m \mid u) = \frac{\exp(F_u \boldsymbol{W}^w F_u^{m\top})}{\sum_{j=1}^{M} \exp(F_u \boldsymbol{W}^w F_u^{j\top})}, \tag{13}$$

where $\boldsymbol{W}^w \in \mathbb{R}^{d_e \times d}$.

We apply another GRU to encode the already generated recommendations $R_{t-1} = \{y_1, y_2, \ldots, y_{t-1}\}$ into $\{\boldsymbol{h}_1^y, \boldsymbol{h}_2^y, \ldots, \boldsymbol{h}_{t-1}^y\}$. Thus, we can then evaluate to what extent each interest is unsatisfied by R_{t-1}, which is denoted as $W(\overline{R_{t-1}}, a_m)$ in Equation (2):

$$\begin{aligned} W(\overline{R_{t-1}}, a_m) &= 1 - \frac{P(a_m \mid u) \exp(w_{t-1}^m)}{\sum_{j=1}^{M} P(a_j \mid u) \exp(w_{t-1}^j)} \\ w_{t-1}^i &= \boldsymbol{W}_{\boldsymbol{y}}^\top \sigma(\boldsymbol{W}_{\boldsymbol{A}} F_u^i + \boldsymbol{W}_{\boldsymbol{B}} \boldsymbol{h}_{t-1}^y), \end{aligned} \tag{14}$$

where w_{t-1}^i indicates the matching between R_{t-1} and F_u^i. The larger value of $W(\overline{R_{t-1}}, a_m)$ indicates that interest a_m is more unsatisfiesd than other interest by the already generated recommendation list and thus should be given more attention when making next recommendation. $P(x \mid a_m)$ in Equation (2) can be calculated by:

$$P(x_n \mid a_m) = \text{softmax}\left(\boldsymbol{x}_{\boldsymbol{n}}^\top F_u^m\right). \tag{15}$$

After selecting the item, with the maximum score based on Equation (2), we can finally add it to the list of recommended items.

3.5 IDP Loss

Different from previous work where trained was based only on cross-entropy loss, we design our loss function considering not only accuracy but also diversity of the generated list of recommended items R_L:

$$\mathcal{L}^{R_L} = \lambda_e \mathcal{L}^{R_L}_{rel} + \mathcal{L}^{R_L}_{div}, \tag{16}$$

where λ_e is a parameter controlling the weight of diversity as well as relevance losses.

3.5.1 Relevance Loss. The relevance loss evaluates if the list of recommended items contains the ground-truth item. Thus, we adopt the conventional cross-entropy loss as:

$$\mathcal{L}^{R_L}_{rel} = -\sum_{i=1}^{|V|} p_i \log\left(q_i^0\right), \tag{17}$$

where R_L is the recommendation list by MDSR, y^* and p_i denotes the ground-truth item, and probability distribution, respectively. q_i^0 indicates the prediction probability conducted by MDSR when generating the first recommendation, where MDSR does not consider the diversity score. In this way, MDSR can also consider the ranking position of the ground-truth item, in the list of recommended items.

3.5.2 Diversity Loss. Since we do not have a ground-truth list for training the model, we employ a self-critic strategy, which can achieve diversity in an unsupervised manner. More specificcally, every step we generate a recommended item, with the scoring function, i.e., Equation (2), and output a list of recommended items R_L, we also generate an item, only with the maximum relevance score $P(v_i \mid u, X_u)$ and yield a list of recommended items R_L^{rel} at the same time. Then we can calculate the diversity loss by

$$\begin{aligned}
\mathcal{L}^{R_L}_{div} &= \boldsymbol{w} \log \frac{1}{1 + \exp(Pr(R_L^{rel}) - Pr(R_L))} \\
Pr(R_L) &= \sum_{v_i \in R_L} \log S(v_i) \\
Pr(R_L^{rel}) &= \sum_{v_i \in R_L^{rel}} \log S(v_i) \\
\boldsymbol{w} &= \boldsymbol{M}(R_L^{rel}) - \boldsymbol{M}(R_L),
\end{aligned} \tag{18}$$

where $Pr(R_L)$ and $Pr(R_L^{rel})$ denotes the log likelihood for generating the lists of recommended items R_L and R_L^{rel}, respectively. $\boldsymbol{w}$ is the gap between the diversity values of the two recommendation list R_L^{rel} and R_L. The diversity value can be calculated by a specific diversity evaluation metric. We adopt **intra-list distance (ILD)** in this article.

The motivation for the design of $\mathcal{L}^{R_L}_{div}$ is the following: We treat R_L^{rel} as a baseline; when we generate a recommendation list that is more diverse than the baseline, we would reward our model to increase the probability of generating R_L; when the generated list R_L is less diverse than R_L^{rel}, we would punish our model with the loss.

In addition to the relevance and diversity losses, two regularization terms are also added to our loss function. Next, we introduce those regularization terms.

3.5.3 Disagreement Regularization. We devise the disagreement regularization to make sure that the multiple interests extracted by the IIM module are different from each other. We consider two strategies.

The first strategy concerns the disagreement on the attention distributions by different interests. We adopt a strategy to enlarge the distance between the attention distributions predicted by each interest. To do so, we use an alignment disagreement regularization [30] as follows:

$$\mathcal{L}_{Dis_{pos}}^{R_L} = \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\left|\boldsymbol{A}^i \odot \boldsymbol{A}^j\right|, \tag{19}$$

where $\boldsymbol{A}^i$ is the attention distribution of interest i. For the multi-head attention-based IIM, $\boldsymbol{A}^i$ can be produced by Equation (6). For the dynamic routing-based IIM, we calculate $\boldsymbol{A}^i$ by doing a softmax operation on similarities between interest capsule i and low-level capsules.

The second strategy is designed to enlarge the dissimilarities between the outputs of multiple interest representations, i.e., F_u^i. We achieve this by minimizing the cosine similarity between each interest F_u^i:

$$\mathcal{L}_{Dis_{out}}^{R_L} = \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\frac{F_u^i \cdot F_u^j}{\left\|F_u^i\right\|\left\|F_u^j\right\|}. \tag{20}$$

3.5.4 Maximum Entropy Regularization. The second regularization is maximum entropy regularization, which can avoid the situation that one of the interests dominates [56, 59]:

$$\mathcal{L}_{ME}^{R_L} = \sum_{m=1}^{M} P(a_m \mid u)\log P(a_m \mid u). \tag{21}$$

Thus, our final IDP loss is as follows:

$$\mathrm{IDP}_{loss}^{R_L} = \lambda_e \mathcal{L}_{rel}^{R_L} + \mathcal{L}_{div}^{R_L} + \mathcal{L}_{Dis}^{R_L} + \mathcal{L}_{ME}^{R_L}. \tag{22}$$

Here, we could add hyper-parameters to control the weights of different regularization terms in our loss. However, this would increase the complexity of optimizing the model due to the additional hyper-parameters' fine-tuning. Thus, for simplicity, we set all weights to 1 and find that different weights would not influence much. All parameters of MDSR can be trained in an end-to-end back-propagation process. The training process of MDSR is shown in Algorithm 2.

For each user sequence X_u, we first initialize the lists of recommendation items R_L and R_L^{rel} as empty sets in step 3 and step 4. Then, we use a GRU as our sequence encoder to generate hidden representations $\boldsymbol{H}_S$ and a global user representation F_u in step 5, step 6 and step 7. Next, we extract the user's multiple interest representations with our IIM module in step 8. Here, we can use either a multi-head attention or dynamic routing-based IIMs. After that, we can generate the first recommendation item, only based on relevance score in step 9 and add it into R_L and R_L^{rel}. In the IDP decoder, we select one item, at a time with a maximum score $S(v)$ and add it R_L in step 18 and step 20. $S(v)$ in step 16 is a combination of relevance as well as diversity scores. At the same time, we also select one item, only with the maximum relevance score $S^{rel}(v)$ and add it to R_L^{rel} in step 19 and step 21, which is then used for calculating the diversity loss $\mathcal{L}_{div}^{R_L}$. Note that we only need to generate R_L^{rel} in the training phase. Finally, we can calculate our designed IDP loss in step 27 based on Equation (22) and use back propagation to optimize the network parameters.

4 EXPERIMENTAL SETUP

We aim to answer the following two research questions with our experiments:

ALGORITHM 2: Training process of the MDSR model.

Input: K: length of recommendation list;
$\boldsymbol{X}$: user behavior sequences set;
V: item, set
Epochs: training iterations;
Output: trainable parameters in MDSR;

for epoch in range (Epochs) **do**
  for $X_u \in X$ **do**
    $R_L = \emptyset$; // initialize R_L as an empty set
    $R_L^{rel} = \emptyset$; // initialize R_L^{rel} as an empty set
    $\boldsymbol{h}_t = \text{GRU}(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t), \boldsymbol{x}_t \in X_u$; // sequence encoder
    $\boldsymbol{H}_S = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_T\}$;
    $F_u = \boldsymbol{h}_T$ // user's global representation;
    F_u^i = multi-head attention($\boldsymbol{H}_S$) or dynamic routing($\boldsymbol{H}_S$); // IIM module
    $v_0 = \arg\max_{v \in V} P(v_n \mid u, X_u)$ with Equation (12) and Equation (13);
    $R_L = R_L \cup v_0$
    $R_L^{rel} = R_L^{rel} \cup v_0$
    for k in range (1, K) **do**
      for $v \in V \setminus R_L$ **do**
        $S^{rel}(v)$ = use Equation (12) and Equation (13)
        $S^{div}(v)$ = use Equation (14) and Equation (15)
        $S(v) = \lambda S^{rel}(v) + (1 - \lambda) S^{div}(v)$;
      end for
      $v_k = \arg\max_{v \in V \setminus R_L} S(v)$ // IDP decoder;
      $v_k^{rel} = \arg\max_{v \in V \setminus R_L} S^{rel}(v)$ // IDP decoder;
      $R_L = R_L \cup v_k$;
      $R_L^{rel} = R_L^{rel} \cup v_k^{rel}$;
    end for
    $\mathcal{L}_{rel}^{R_L} = -\sum_{i=1}^{|V|} p_i \log\left(q_i^0\right)$ // relevance loss
    $\mathcal{L}_{div}^{R_L} = \boldsymbol{w} \log \frac{1}{1+\exp(Pr(R_L^{rel}) - Pr(R_L))}$ // diversity loss
    $\mathcal{L}_{Dis}^{R_L} = \mathcal{L}_{Dis_{pos}}^{R_L}$ or $\mathcal{L}_{Dis_{out}}^{R_L}$ // disagreement regularization
    $\mathcal{L}_{ME}^{R_L} = \sum_{m=1}^{M} P(a_m \mid u) \log P(a_m \mid u)$ // maximun entropy regularization
    $\text{IDP}_{loss}^{R_L} = \lambda_e \mathcal{L}_{rel}^{R_L} + \mathcal{L}_{div}^{R_L} + \mathcal{L}_{Dis}^{R_L} + \mathcal{L}_{ME}^{R_L}$; // IDP loss
    use back propagation to optimize the parameters
  end for
end for
return parameters in MDSR.

(**RQ1**) Can MDSR beat the state-of-the-art baselines in terms of accuracy?
(**RQ2**) Does MDSR outperform state-of-the-art baselines regarding diversity?

4.1 Datasets

We conduct our experiments on four datasets, i.e., two movie recommendation datasets and two e-commerce datasets. We report detailed statistics of those datasets in Table 2.

Table 2. Dataset Statistics

Dataset	ML100K	ML1M	Tafeng	Tmall
Number of users	943	6,022	1,703	25,958
Number of items	1,349	3,043	2,461	57,677
Number of interactions	93,629	959,022	42,921	623,124
Number of item, categories	19	18	469	70
Avg. number of genres per item,	1.7	1.6	1.0	1.0

- **ML100K**[1] and **ML1M**[1] are both collected from the MovieLens web site. ML1M is much larger and sparser than ML100K.
- **Tafeng**[2] contains one-month user shopping logs in a grocery store.
- **Tmall**[3] is collected by an online shopping website, Tmall, which also includes users' shopping behavior.

As for the category information, it needs to be pointed out that a movie in the ML1M and ML100K datasets may belong to several genres while a single item, in the Tafeng and Tmall datasets is only related to one category at the same time.

We process the data following Li et al. [29]. In ML100K, users with fewer than 5 interactions and items that are clicked less than 5 times are filtered out. For other datasets, users and items that have fewer than 20 interactions are filtered out; to keep the sequential characteristics of the datasets, we sort user behavior according to the "timestamp" field. To generate the sequences, we adopt a sliding-window strategy that uses the past 9 interactions as input and predicts the 10th interaction. The first 80% of each dataset is used for training the model while the last 10% is for testing and the remaining 10% is for validation. We also ensure that the items in test set have been interacted with by at least one user in the training set.

4.2 Methods Used for Comparison

Diversity in recommendations can be divided into two types: individual diversity and aggregate diversity [2, 37]. The individual diversity is a measure of average dissimilarity of items recommended to an individual user while the aggregate diversity is the total number of distinct items recommended across all users [37]. As the authors of S-DIV [25] point out, S-DIV focuses on the aggregate diversity and recommends general as well as tail items. However, in our article the main concern is individual diversity. Thus, we do not compare with S-DIV. What's more, S-DIV applies the content information of an item, which is not used in our model. The item, content is beneficial for diversity, because by mapping items into a content (latent) space, the items are not limited to specific item, IDs but generalized to certain broad contexts with high variability. We would like to incorporate the content information in our future work. Besides, in this article we aim to improve accuracy as well as diversity for SR in an end-to-end framework. Thus, we do not compare with models that only focus on improving accuracy for SR. For example, BERT4Rec [46] and SASRec [24] are recently proposed methods for SR that can also be integrated into our sequence encoder component to help improve the recommendation accuracy.

We choose state-of-the-art neural SR methods that share a similar structure as ours as baselines to compare with in Table 3.

- **GRU4Rec**: GRU4Rec is the first work that introduces RNNs into SR and proposes a session-parallel training mechanism [17].

[1] https://grouplens.org/datasets/movielens/.
[2] https://www.kaggle.com/chiranjivdas09/ta-feng-grocery-dataset.
[3] https://tianchi.aliyun.com/dataset/dataDetail?dataId=42.

Table 3. An Overview of the Models Discussed in the Article

Model	Description	Source
GRU4Rec	The first work that introduces RNNs into SR and propose a session-parallel training mechanism.	[17]
NARM	A sequential recommendation model that applies an attention mechanism upon an RNN.	[29]
MCPRN	A method that captures users' multiple intents in a session with mixture-channel recurrent networks.	[51]
NARM+MMR	A re-ranking model for generating diversified recommendations for SR. It applies MMR to re-rank the candidate items generated by NARM.	[10]
IDSR	Our previously proposed method that improves sequential recommendation diversification in an end-to-end framework.	[10]
MDSR_{MA}	MDSR model using *Multi-head Attention* method in IIM module to extract users' multiple interests.	This article.
MDSR_{DR}	MDSR model using *Dynamic Routing* method in IIM module to extract users' multiple interests.	This article.

- **NARM**: A sequential recommendation model that applies an attention mechanism on top of an RNN. It combines the attentive hidden states and the final hidden state to represent a user's current preference and thus makes accurate recommendations [29].
- **MCPRN**: MCPRN models users' multiple interests in a session with mixture-channel recurrent networks [51]. It has been shown that MCPRN is able to improve the performance regarding both accuracy and diversity.
- **NARM+MMR**: Besides the baselines obtained from prior work listed above, we also construct a baseline ourselves, i.e., **NARM+MMR**, which reranks the predicted list of NARM in a post-processing step. More precisely, NARM first generates a candidate recommendation set, i.e., R_c, and each item, in the set is accompanied with its relevance score $S(v_i)$ predicted by NARM. Then, MMR is applied to rerank the items in the candidate set so as to generate a diversified recommendation list. The criteria used in MMR to sort these items is as follows:

$$v \leftarrow \underset{v_i \in R_c \setminus R_L}{\arg\max}\, \theta S(v_i) + (1-\theta) \min_{v_k \in R_L} d_{ki},$$

  where d_{ki} is the distance between item, v_k and item, v_i; $\theta \in [0, 1]$ is a hyper-parameter that needs to be carefully tuned. We first initialize the recommendation list R_L as an empty set and then select one item, based on Equation (23) and add it to R_L at every step iteratively until $|R_L| = N$. The hyper-parameter controls the diversity of the final recommendation list, e.g., when $\theta = 1$, the output list R_L is the same as the one generated by NARM; when $\theta = 0$, the output list R_L is aimed to maximize its diversity and ignore relevance.
- **IDSR**: This is our own recent work [10], which models users' multiple intents and aims to improve recommendation diversity in an end-to-end structure for SR.

As to our proposed model MDSR in this article, we consider two variants:

- **MDSR_{MA}:** MDSR_{MA} is a variant of MDSR that uses a *multi-head attention* method in the IIM module to extract users' multiple interests.
- **MDSR_{DR}:** MDSR_{DR} is a variant of MDSR that uses a *dynamic routing* method in the IIM module to extract users' multiple interests.

4.3 Evaluation Metrics

For accuracy, we apply Recall and MRR following [29, 33]. For diversity, intra-list distance (ILD) [58] is used as the evaluation metric, which is regularly applied in recommendation diversification task.

- **Recall:** The value is determined by whether the ground-truth item, is included in the list of recommended items. If the ground-truth item, is in the list, then the value is 1; otherwise, it equals 0.
- **MRR:** The value of determined by the position of the ground-truth item, in the list of recommended items. The higher the position, the larger the value.
- **ILD:** The value is determined by the average distance between pairs of items in the recommendation list. It can be formulated as

$$\mathrm{ILD} = \frac{2}{|R_L|(|R_L|-1)} \sum_{(i,j)\in R_L} d_{ij}. \tag{23}$$

The dissimilarity d_{ij} between two items is calculated as Euclidean distance between the item, genre vectors [4].

4.4 Implementation Details

For the hyper-parameters of the MDSR model, the size of the item, embedding is set to 128; the GRU hidden state size is 128; the parameter λ_e is set to 1.0 for the two movie datasets and the Tmall datasets, and 0.1 for the Tafeng dataset. For the training process, we set the mini-batch size to 512, use the Xavier method [14] to initialize the model parameters, and optimize the model using Adam [26] where the initial learning rate $\alpha = 0.001$, two momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. For validation, we show the model performance for every epoch on the validation set and save the model after it achieves the best performance. For testing, we show the results on the test set with the saved model. In this article, we consider top-10 recommendation, i.e., $K = 10$. We put the code for realizing our model online.[4]

5 RESULTS

5.1 Performance in Terms of Accuracy

For RQ1, we compare MDSR with the models listed in Table 3 in terms of accuracy. The results are listed in Table 4.

First, since the NARM has a similar encoding structure as MDSR, we can see that the performance of MDSR_{MA} and MDSR_{DR} is comparable or even superior to NARM regarding recommendation accuracy, i.e., Recall and MRR. Although MDSR tries to diversify the recommendations, at the first generating steps, it still assigns high probability to these relevant items. Additionally, the IDP loss also considers recommendation accuracy, which helps MDSR to capture users' major interests. When users have multiple interests, as NARM shows bias toward the main interest, the recommendations may be unsatisfactory to those users. For example, MDSR_{MA} outperforms NARM on the Tafeng dataset, where the improvements in terms of Recall and MRR are 10.19% and 10.06%, respectively. We think that this may be because that Tafeng collects users' behavior in a grocery store, where users typically have multiple interests when shopping, and buy items with different categories. It also can be found that MDSR_{MA} and MDSR_{DR} show better performance in terms of accuracy on all of the four datasets than MCPRN. Compared with MCPRN, the IIM

[4] https://bitbucket.org/WanyuChen/idsr/src/master/.

Table 4. Performance of All Discussed Models in This Article

Dataset	Metric	GRU4Rec	NARM	MCPRN	NARM+MMR	IDSR	MDSR$_{MA}$	MDSR$_{DR}$
ML100K	Recall (%)	6.23	<u>9.68</u>	9.27	9.53	9.79	**9.88**	9.62
	MRR (%)	2.09	<u>3.18</u>	2.99	2.77	3.22	**3.53**$^{\triangle}$	3.35
	ILD	1.527	1.518	1.561	<u>1.583</u>	1.666	1.646	**1.669**$^{\triangle}$
ML1M	Recall (%)	11.67	<u>15.02</u>	14.89	14.72	14.89	**15.72**	15.18
	MRR (%)	4.02	<u>5.39</u>	5.26	4.89	5.30	**6.23**$^{\triangle}$	5.97
	ILD	1.307	1.289	1.301	<u>1.325</u>	1.383	1.486	**1.500**$^{\triangle}$
Tafeng	Recall (%)	4.11	<u>4.71</u>	4.57	4.33	4.97	**5.19**$^{\triangle}$	4.94
	MRR (%)	1.42	<u>1.69</u>	1.6	1.41	**1.96**$^{\triangle}$	1.86	1.80
	ILD	1.267	1.214	1.248	<u>1.263</u>	1.318	**1.364**$^{\triangle}$	1.344
Tmall	Recall (%)	12.11	<u>14.41</u>	14.19	14.00	14.32	14.78	**14.89**
	MRR (%)	5.41	<u>7.51</u>	7.27	6.28	7.43	7.86	**7.95**$^{\triangle}$
	ILD	0.879	0.834	0.886	<u>0.892</u>	**0.946**$^{\triangle}$	0.926	0.936

The results of the best baseline and the best model in each row are underlined and in bold, respectively. Statistical significance of pairwise differences of best model vs. the best baseline is determined by a paired t-test ($^{\triangle}$ for p-value ≤ 0.05).

module in MDSR cares not only users' multiple interests but also the weight of each interest, which is effective in improving the recommendation accuracy.

Second, it is obvious that after re-ranking, the accuracy of NARM drops dramatically when we compare the performance of NARM and NARM+MMR. This demonstrates that the reranking algorithm, i.e., MMR, hurts the accuracy of recommendation although it can improve the diversity to some extent. This may be because the candidate items recommended by NARM often share similar genres or categories. When the diversity scores for the relevant items are lower than the irrelevant items, the irrelevant items will have higher scores than these relevant ones, which thus results in a worse accuracy performance. Moreover, the reranking algorithm, i.e., MMR, is computationally expensive, since it needs to compare each pair of items in the generated list of recommendations and the candidate set.

MDSR$_{MA}$ has the same structure as IDSR except that MDSR$_{MA}$ uses layer normalization and removes the feedforward layer. However, MDSR$_{MA}$ shows better performance than IDSR in terms of MRR and Recall on most of our datasets, which demonstrates the effectiveness of applying the layer normalization strategy. MDSR$_{MA}$ is better at capturing users' main interests than MDSR$_{DR}$. For instance, MDSR$_{MA}$ shows better performance than MDSR$_{DR}$ on the ML100K, ML1M and Tafeng datasets in terms of Recall and MRR. We will give a detailed analysis of the performance of the two variants of the MDSR model in Section 6.1.

In summary, MDSR shows comparable or superior performance as the baseline models in terms of recommendation accuracy. It should be noticed that we can add other effective structures into the MDSR framework to improve the recommendation accuracy, e.g., SASRec [24]. However, this can be explored as our future work.

5.2 Performance in Terms of Diversity

For RQ2, we show the diversity scores, i.e., ILD, on all of the four datasets in Table 4. We see that MDSR outperforms all baselines. Specifically, the improvements of MDSR$_{DR}$ over mixture-channel purpose routing network (MCPRN) are 6.92%, 15.30%, and 5.64% in terms of ILD on the ML100K, ML1M, and Tmall datasets, respectively, while MDSR$_{MA}$ beats MCPRN with a 9.29% improvement on the Tafeng dataset. Although MCPRN models users' multiple interests, there is no supervision signal to help the model learn to discriminate those multiple interests and produce diverse recommendations. However, in MDSR, the diversity loss and disagreement regularization term in the IDP loss can guide the model to learn to distinguish different interests and satisfy them gradually.

MDSR significantly outperforms NARM+MMR. For example, the improvements of MDSR_{DR} over NARM+MMR are 5.43% and 13.21% on ML100K and ML1M, respectively. We can also find that MMR depends heavily on the recommendation results of NARM. When the items in the candidate set generated by NARM show similar categories, the reranking performance of MMR method is limited. However, MDSR can avoid this problem, since it can learn to diversify the recommendations by optimizing the designed IDP loss in Equation (22).

MDSR_{DR} outperforms IDSR on the ML100K, ML1M, and Tafeng datasets in terms of ILD while it loses by 1% on the Tmall dataset, which is not significant. Comparing MDSR_{MA} and MDSR_{DR}, we can see that MDSR_{DR} also shows better performance than MDSR_{MA} in terms of ILD on most datasets. Dynamic routing acts like a soft-clustering algorithm: It can group a user's sequential behaviors into several clusters with each cluster representing a certain latent interest of the user. In this way, it can help to distinguish between users' different interests.

6 ANALYSIS

In this section, we conduct several experiments to analyze the performance of MDSR in more depth. Specifically, we seek to answer the following questions mainly:

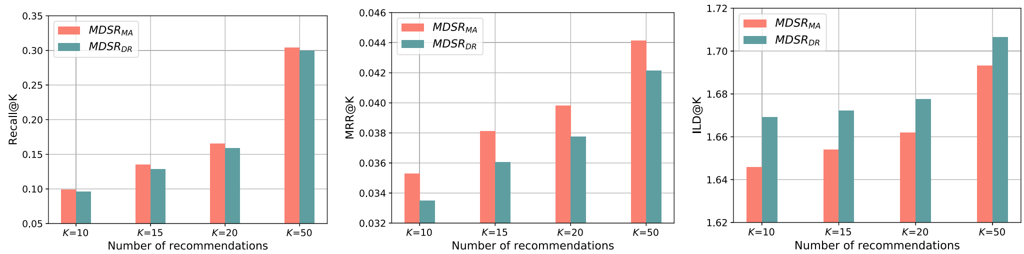
- What is the performance of the two strategies in the IIM module, i.e., *multi-head attention* and *dynamic routing*?
- What is the influence of the number of latent interests in MDSR?
- How does the tradeoff parameter λ influence the performance of MDSR?
- What is the effect of different disagreement regularization terms $\mathcal{L}_{Dis}^{R_L}$ in Equation (22), i.e., $\mathcal{L}_{Dis_{pos}}^{R_L}$ and $\mathcal{L}_{Dis_{out}}^{R_L}$?
- What is the effect of the maximum entropy regularization loss $\mathcal{L}_{ME}^{R_L}$ in Equation (22)?
- Does the IIM module in MDSR capture users' multiple interests?

6.1 Impact of Different Multiple Interest Extractors

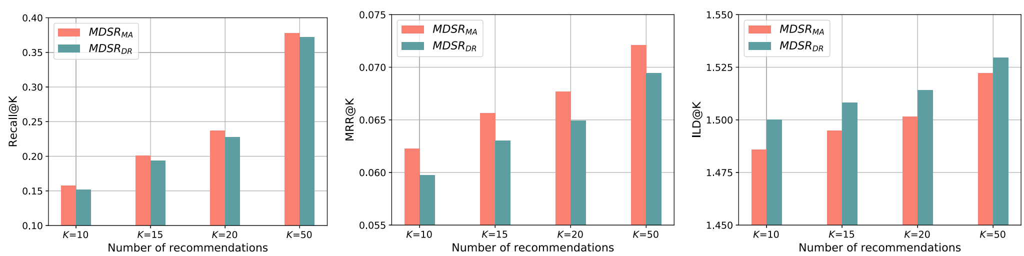
We visualize the performance of MDSR with two strategies for the IIM module for top K recommendation (with $K = 10, 15, 20, 50$) in Figure 3.

We see that with the increases in K, i.e., from 10 to 50, the performance of both MDSR_{MA} and MDSR_{DR} improves in terms of accuracy and diversity. Specifically, on the MovieLens datasets shown in Figure 3(a) (ML100K) and Figure 3(b) (ML1M), MDSR_{MA} shows better performance than MDSR_{DR} in terms of accuracy but a worse performance than MDSR_{DR} in terms of diversity. It should be noted that the gap between the two models in terms of MRR is larger than in terms of ILD. For example, when $K = 10$ and $K = 50$, the improvements of MDSR_{MA} over MDSR_{DR} are 5.37% and 4.71% in terms of MRR, respectively, while the improvements of MDSR_{DR} over MDSR_{MA} are 1.43% and 0.79% in terms of ILD, neither of which is significant. This demonstrates that the multi-head attention mechanism is better at capturing users' main interests.

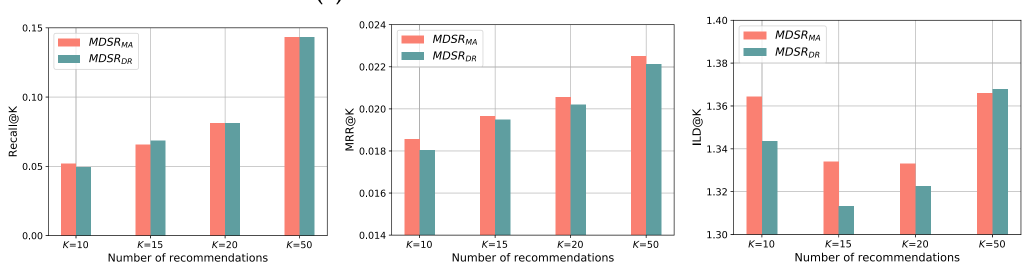
However, on the e-commerce datasets shown in Figure 3(c) (Tafeng) and Figure 3(d) (Tmall), MDSR_{MA} and MDSR_{DR} have comparable performance in terms of accuracy as well as diversity. This is because users tend to have multiple interests and purchase items from different categories in e-commerce instead of having a main purpose in mind. This finding is in accordance with the analysis in Section 5.1.



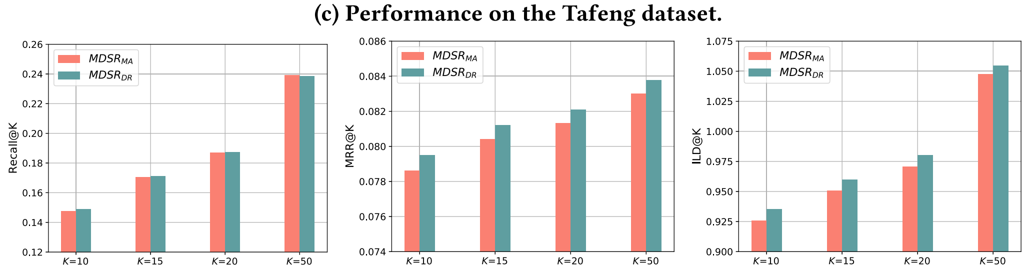
(a) Performance on the ML100K dataset.



(b) Performance on the ML1M dataset.



(c) Performance on the Tafeng dataset.



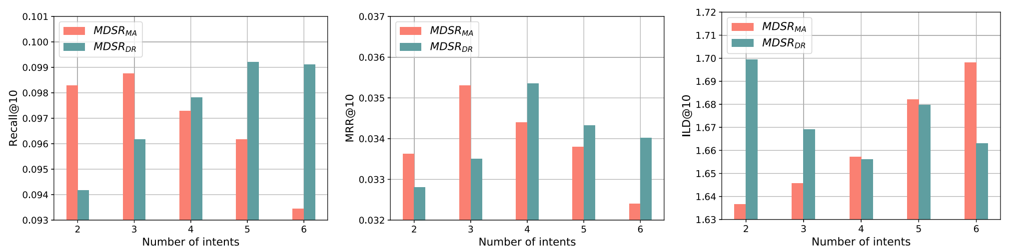
(d) Performance on the Tmall dataset.

Fig. 3. Performance of MDSR with different interest extractors on the top K recommendation task, with $K = 10, 15, 20, 50$, on four datasets.

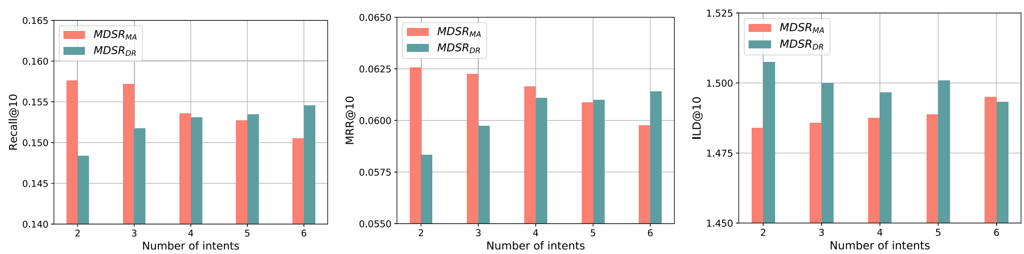
6.2 Impact of the Number of Latent Interests

Next, we show the performance of MDSR when the number of latent interests ranges from 2 to 6 in Figure 4.

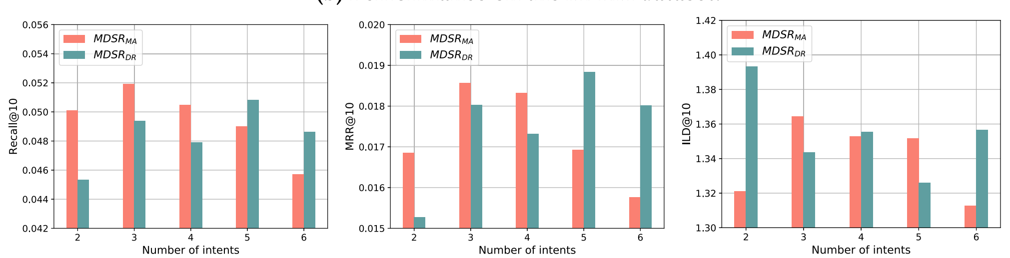
We can see that both MDSR$_{MA}$ and MDSR$_{DR}$ show upward trends in accuracy when the number of interests increases from 2 to 3, except for MDSR$_{MA}$ on the Tmall dataset. MDSR$_{MA}$ achieves its best performance in terms of accuracy with 3 interests and then begins to decrease when the



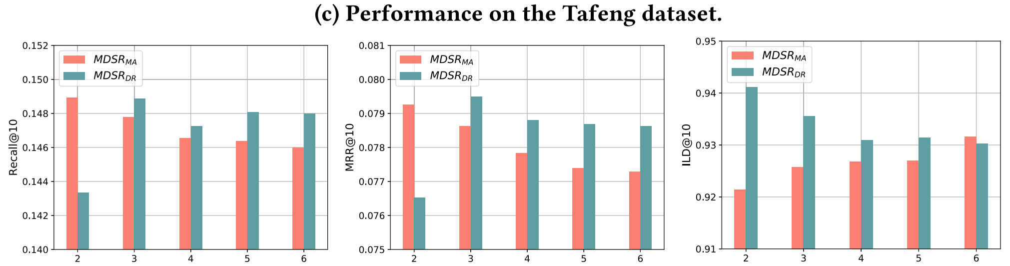
(a) Performance on the ML100K dataset



(b) Performance on the ML1M dataset.



(c) Performance on the Tafeng dataset.



(d) Performance on the Tmall dataset.

Fig. 4. Performance of the MDSR models on four datasets with different numbers of latent interests.

number of interests increases from 3 to 6 on the ML100K, ML1M, and Tafeng datasets. On the Tmall dataset, when increasing the number of interests, the performance of MDSR$_{MA}$ gets worse constantly. MDSR$_{DR}$ achieves its best performance in terms of accuracy on the ML100K, ML1M, and Tmall datasets when the number of interests is 3 or 4. But on the Tafeng dataset, its best performance is achieved when the number of interests is 5. This might be related to the fact that the Tafeng dataset has considerably more categories than the other datasets, i.e., 467 vs. 17, 18 and 70. Users are likely to have more interests especially on the dataset with more categories of items, thus modeling multiple interests can help to increase the recommendation accuracy.

The diversity score achieved by MDSR_{MA} increases slightly when assuming more interests on the ML100K, ML1M, and Tmall datasets. But, on the Tafeng dataset, it achieves its best performance with 3 interests, which is consistent with its accuracy performance. As for MDSR_{DR}, the number of interests has little influence on its performance in terms of ILD, which demonstrates that it has better stability than MDSR_{MA} in terms of recommendation diversification.

In general, Figure 4 shows that increasing the number of interests will hurt accuracy. Therefore, we choose to set the number of latent interests to three in our experiments that are tuned on the validation set.

6.3 Influence of the Tradeoff Parameter λ

To see the impact of the tradeoff parameter λ on MDSR, we show the results of MDSR_{MA} and MDSR_{DR} on all datasets by ranging λ from 0 to 1 with a step size of 0.1. The results are plotted in Figure 5.

Both models show upward trends generally when λ increases from 0 to 1 on the MovieLens datasets in terms of Recall and MRR. When $\lambda = 0$, MDSR shows the worst performance. An obvious increase is observed when λ increases from 0 to 0.1. $\lambda = 0$ denotes that we focus on diversity and ignore accuracy, thus the accuracy is worst. Similar trends can be observed on the e-commerce datasets regarding MRR and Recall, except that MDSR_{DR} drops slightly when λ increases from 0.9 to 1 on the Tafeng dataset. This indicates that considering diversity can also help to improve the accuracy performance, especially on a dataset with a large number of categories. Since users tend to have more kinds of interests when faced with more categories of items, diversified recommendations may increase the probability of containing items that users want.

Both models achieve the best performance in terms of ILD when $\lambda = 0.0$ on all of the four datasets as we consider the recommendation diversification only. When λ increases from 0 to 1, the ILD scores drops on all datasets. There are more fluctuations on the e-commerce datasets than on the MovieLens datasets, especially on Tafeng. The performance of MDSR in terms of ILD drops sharply from 0 to 0.1.

To sum up, increasing the value of λ will improve the performance in terms of accuracy while hurting diversity, especially when it changes from 0 to 0.1. We set it to 0.2 on the ML100K, ML1M and Tafeng datasets and to 0.5 on the Tmall dataset for overall performance evaluation.

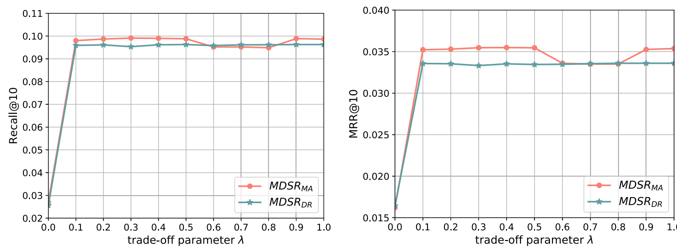
6.4 Effect of Different Disagreement Regularizations

To analyze the influence of the disagreement regularization losses, i.e., $\mathcal{L}^{R_L}_{Dis_{pos}}$ and $\mathcal{L}^{R_L}_{Dis_{out}}$, we compare the following IDP loss variants:

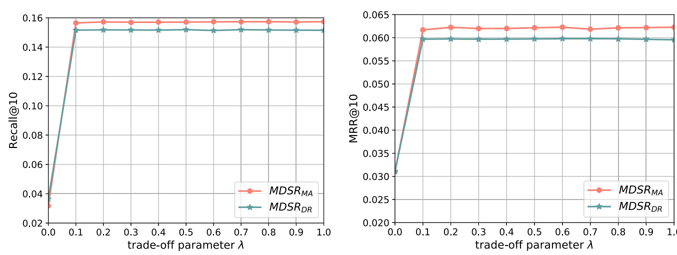
$$\begin{aligned}
\mathcal{L}_0 &= \lambda_e \mathcal{L}^{R_L}_{rel} + \mathcal{L}^{R_L}_{div} + \mathcal{L}^{R_L}_{ME} \\
\mathcal{L}_{out} &= \lambda_e \mathcal{L}^{R_L}_{rel} + \mathcal{L}^{R_L}_{div} + \mathcal{L}^{R_L}_{Dis_{out}} + \mathcal{L}^{R_L}_{ME} \\
\mathcal{L}_{pos} &= \lambda_e \mathcal{L}^{R_L}_{rel} + \mathcal{L}^{R_L}_{div} + \mathcal{L}^{R_L}_{Dis_{pos}} + \mathcal{L}^{R_L}_{ME},
\end{aligned} \tag{24}$$

where $\mathcal{L}_0$, $\mathcal{L}_{out}$, and $\mathcal{L}_{pos}$ denote the IDP loss without disagreement regularization, with output disagreement regularization, and with position disagreement regularization, respectively. The results are shown in Table 5.

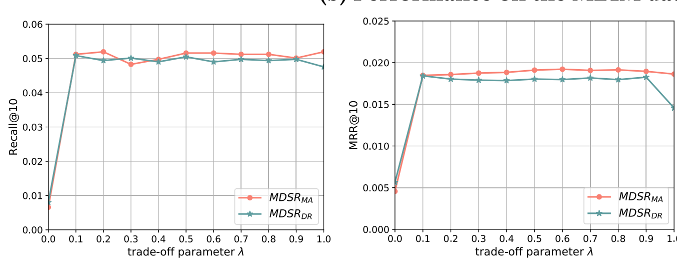
Comparing $\mathcal{L}_0$ with $\mathcal{L}_{out}$ or $\mathcal{L}_{pos}$ on all datasets, we can see that, in general, $\mathcal{L}^{R_L}_{Dis}$ can help to boost the performance of MDSR in terms of diversity. This indicates that the IIM module can capture multiple latent interests effectively by applying the $\mathcal{L}^{R_L}_{Dis}$ loss. Specifically, MDSR_{MA} with the $\mathcal{L}_{pos}$ loss always shows better performance than with the $\mathcal{L}_{out}$ loss in terms of ILD. It



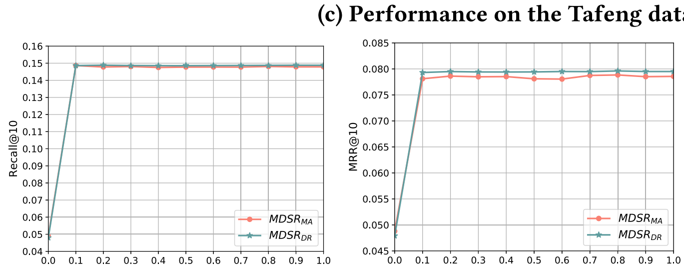
(a) Performance on the ML100K dataset.



(b) Performance on the ML1M dataset.



(c) Performance on the Tafeng dataset.



(d) Performance on the Tmall dataset.

Fig. 5. Performance of the MDSR models on four datasets where the parameter λ in Equation (2) ranges from 0 to 1 with a step size of 0.1.

indicates that, as for the multi-head attention mechanism, the disagreement regularization can help to distinguish those interests more effectively when applied to positions than to outputs.

As for MDSR$_{DR}$, the dynamic routing algorithm plays a role similar to the clustering algorithm, which can distinguish users' different interests intrinsically. Thus even without $\mathcal{L}_{Dis}^{R_L}$, MDSR$_{DR}$ still shows better performance than MDSR$_{MA}$ with $\mathcal{L}_0$ and with $\mathcal{L}_{out}$ on most datasets in terms of ILD. But, we can see that there are still small improvements in terms of ILD after applying $\mathcal{L}_{out}$ or $\mathcal{L}_{pos}$ to MDSR$_{DR}$. This is because the dynamic routing algorithm is similar to the k-means

Table 5. Performance of MDSR on Four Datasets with/without Different Types of Disagreement Regularization

Metric	Model	$\mathcal{L}_0$	$\mathcal{L}_{out}$	$\mathcal{L}_{pos}$
Recall (%)	MDSR$_{MA}$	10.32	10.19	9.87
	MDSR$_{DR}$	10.01	9.61	9.63
MRR (%)	MDSR$_{MA}$	3.56	3.50	3.53
	MDSR$_{DR}$	3.49	3.35	3.38
ILD	MDSR$_{MA}$	1.58	1.61	1.65
	MDSR$_{DR}$	1.63	1.67	1.64

(a) ML100K

Metric	Model	$\mathcal{L}_0$	$\mathcal{L}_{out}$	$\mathcal{L}_{pos}$
Recall (%)	MDSR$_{MA}$	16.10	15.67	15.72
	MDSR$_{DR}$	15.92	15.17	13.26
MRR (%)	MDSR$_{MA}$	6.36	6.21	6.22
	MDSR$_{DR}$	6.26	5.97	5.22
ILD	MDSR$_{MA}$	1.32	1.46	1.48
	MDSR$_{DR}$	1.48	1.50	1.50

(b) ML1M

Metric	Model	$\mathcal{L}_0$	$\mathcal{L}_{out}$	$\mathcal{L}_{pos}$
Recall (%)	MDSR$_{MA}$	5.61	5.19	5.19
	MDSR$_{DR}$	4.83	4.94	4.79
MRR (%)	MDSR$_{MA}$	2.06	1.79	1.86
	MDSR$_{DR}$	1.93	1.80	1.82
ILD	MDSR$_{MA}$	1.27	1.34	1.36
	MDSR$_{DR}$	1.30	1.34	1.32

(c) Tafeng

Metric	Model	$\mathcal{L}_0$	$\mathcal{L}_{out}$	$\mathcal{L}_{pos}$
Recall (%)	MDSR$_{MA}$	15.05	14.99	14.78
	MDSR$_{DR}$	15.05	14.88	14.89
MRR (%)	MDSR$_{MA}$	8.04	7.95	7.86
	MDSR$_{DR}$	8.01	7.95	8.00
ILD	MDSR$_{MA}$	0.82	0.88	0.93
	MDSR$_{DR}$	0.90	0.94	0.91

(d) Tmall

algorithm [15], which is an unsupervised learning method. It aims to narrow down the distance among items within the same cluster in an iterative way. In most of the experiments with dynamic routing, the number of iterations is often set to 3 (we also set it to 3 in our experiments) [28]. However, the dynamic routing algorithm may not be able to converge to a good clustering result within this number of iterations. In this case, the disagreement regularization can provide assistance to the dynamic routing algorithm by enlarging the distance between interests representations. Since the number of latent interests is small, it would not cost too much calculation when applying the disagreement regularization.

6.5 Effect of Maximum Entropy Regularization

To analyze the influence of the maximum entropy regularization $\mathcal{L}_{ME}^{R_L}$ on MDSR, we consider the following IDP loss variant for comparison:

$$\mathrm{IDP}_{loss}^{R_L} = \lambda_e \mathcal{L}_{rel}^{R_L} + \mathcal{L}_{div}^{R_L} + \mathcal{L}_{Dis}^{R_L} + \lambda_{ME}\mathcal{L}_{ME}^{R_L}, \tag{25}$$

where $\mathcal{L}_{ME}^{R_L}$ is weighted by the parameter λ_{ME}. We test the performance of MDSR variants with λ_{ME} = 0.0, 0.5, and 1.0, respectively. The results are reported in Table 6.

We can see that $\mathcal{L}_{ME}^{R_L}$ can help improve the performance of MDSR$_{MA}$ in terms of both accuracy and diversity on all datasets. Specifically, on the e-commerce datasets, the performance of MDSR$_{MA}$ gets more improvements on diversity than on accuracy. For example, when comparing λ_{ME}=1.0 with λ_{ME}=0.0, MDSR$_{MA}$ shows 5.43% improvements in terms of ILD while 1.36% improvements in terms of Recall on the Tafeng dataset. On the Tmall dataset, the improvements are 4.49% and 0.89% in terms of ILD and Recall, respectively. This demonstrates that the maximum entropy regularization can alleviate the issue that one interest dominates and hereby can help to improve the diversification performance.

Similar results can be found for MDSR$_{DR}$ except that, on the ML100K dataset, $\mathcal{L}_{ME}^{R_L}$ leads to a slight drop in the accuracy performance. This might be because the maximum entropy regularization forces the model to treat multiple interests equally. However, users tend to have one main interest when choosing movies to watch next, thus MDSR$_{DR}$ cannot capture the main interest well with the maximum entropy regularization. Additionally, the time gap between adjacent

Table 6. Performance of MDSR on Four Datasets with Different Weights of Maximum Entropy Regularization

Metric	Model	λ_{ME} = 0.0	λ_{ME} = 0.5	λ_{ME} = 1.0
Recall (%)	MDSR$_{MA}$	9.61	9.85	9.92
	MDSR$_{DR}$	9.92	9.64	9.42
MRR (%)	MDSR$_{MA}$	3.39	3.51	3.54
	MDSR$_{DR}$	3.41	3.38	3.29
ILD	MDSR$_{MA}$	1.61	1.65	1.64
	MDSR$_{DR}$	1.62	1.67	1.65

(a) ML100K

Metric	Model	λ_{ME} = 0.0	λ_{ME} = 0.5	λ_{ME} = 1.0
Recall (%)	MDSR$_{MA}$	13.94	15.68	15.64
	MDSR$_{DR}$	15.00	15.14	15.17
MRR (%)	MDSR$_{MA}$	5.33	6.21	6.28
	MDSR$_{DR}$	5.98	5.98	6.02
ILD	MDSR$_{MA}$	1.41	1.49	1.48
	MDSR$_{DR}$	1.44	1.50	1.49

(b) ML1M

Metric	Model	λ_{ME} = 0.0	λ_{ME} = 0.5	λ_{ME} = 1.0
Recall (%)	MDSR$_{MA}$	5.12	5.19	5.19
	MDSR$_{DR}$	4.97	4.97	4.93
MRR (%)	MDSR$_{MA}$	1.88	1.85	1.89
	MDSR$_{DR}$	1.65	1.80	1.73
ILD	MDSR$_{MA}$	1.29	1.36	1.36
	MDSR$_{DR}$	1.30	1.34	1.36

(c) Tafeng

Metric	Model	λ_{ME} = 0.0	λ_{ME} = 0.5	λ_{ME} = 1.0
Recall (%)	MDSR$_{MA}$	14.58	14.71	14.85
	MDSR$_{DR}$	14.72	14.87	14.93
MRR (%)	MDSR$_{MA}$	7.73	7.83	7.84
	MDSR$_{DR}$	7.83	7.93	7.94
ILD	MDSR$_{MA}$	0.89	0.93	0.93
	MDSR$_{DR}$	0.90	0.93	0.93

(d) Tmall

Fig. 6. An example of movies recommended by the MDSR models and NARM.

interactions on the ML100K dataset is larger than that on Tafeng and Tmall, so previous behavior and multiple interests have less influence on users' current interactions on the ML100K dataset.

6.6 Case Study

In this subsection, we give a case study from the test set of the ML100K dataset to show the recommendations generated by different MDSR variants; see Figure 6. The first sequence in Figure 6 shows 9 movies that the user has watched, and the next two lists show the top 10 recommendations generated by MDSR$_{DR}$ and MDSR$_{MA}$, respectively. The last list shows the top 10 recommendations by NARM. The ground-truth item, is marked in a red box.

Table 7. Complexity Analysis

Model	Computational complexity
MDSR$_{MA}$	$O(nd^2 + Mnd^2 + Mnd + Nd^2 + NMd + NMd^2)$
MDSR$_{DR}$	$O(nd^2 + RMnd^2 + RM^2nd + Nd^2 + NMd + NMd^2)$
NARM	$O(nd^2 + nd^2 + nd)$

Based on the user's historical interactions, we see that, recently, the user mostly favors movies with Action, Adventure, Sci-Fi, and Children genres. But the user also shows interest in Comedy, Thriller, Musical, and Romance. The items recommended by neural attentive recommendation machine (NARM) mainly belong to the Action, Adventure and Sci-Fi genres, e.g., most of them are action movies, which is close to the recent interest of this user. But NARM ignores other genres that are also reflected in the user's watching behavior, e.g., the ones with red boxes in Figure 6. In contrast, both MDSR variants accommodate multiple interests and diversify the list of recommended movies with Action, Adventure, Sci-Fi, Children, Comedy, Thriller, Musical, and Romance genres. This indicates that MDSR cannot only extract users' multiple interests, but also produce diversified recommendations to cover and satisfy those interests. Besides, MDSR also identifies the most important interest and assigns a high probability to the right one. For example, the MDSR$_{DR}$ model returns the ground-truth item, at the first position in the recommendation list; for the MDSR$_{MA}$ model, the ground-truth item, is ranked at the second position in the recommendation list; for the baseline model NARM, the ground-truth item, is ranked at the first position in the recommendation list. Thus it also demonstrates that MDSR can achieve the competitive performance in terms of recommendation accuracy with the baseline model.

6.7 Computational Complexity Analysis

In this section, we give a brief analysis of the computational complexity of our proposed MDSR models with multi-head attention mechanism and dynamic routing algorithm as well as the baseline model NARM in Table 7, respectively. For convenience, we make the following settings. The dimensions of item, embeddings and hidden layers are set to d, the length of a user's behavior sequence is n, the number of latent interests is M, the number of recommendations is N, and the number of iterations for dynamic routing algorithm is R. As $M \ll d$, $N \ll d$ and $R \ll d$, we can reduce some parts of the calculation.

The computation complexity of the sequence encoder module is: nd^2; the Multi-head attention-based IIM module is $Mnd^2 + Mnd$; the Dynamic routing-based IIM module is $RMnd^2 + RM^2nd$; the IDP decoder is $Nd^2 + NMd + NMd^2$. It should be noted that except the GRU and the Dynamic routing related components that need to be executed in a recurrent way, other components in the IDP decoder and the Multi-head attention-based IIM module can be computed in a parallel way. Comparing the computational complexity of the three models, we can find that the complexity of MDSR is higher than NARM, which mainly comes from the IDP decoder module. We may focus on this component and try to reduce the complexity by using self-attention mechanism instead of GRU or applying some efficiency improving techniques in our future work.

7 CONCLUSION AND FUTURE WORK

In this article, we have regarded the sequential recommendation task as a list generation process and proposed the MDSR model. We have introduced an IIM module to extract users' multiple interests and an IDP decoder to produce diversified recommendations satisfying those interests gradually. We have designed an IDP loss to supervise the model to consider accuracy and diversification simultaneously during training.

Comprehensive experiments have been conducted on four datasets. The results have shown that MDSR significantly outperforms the state-of-the-art baselines regarding recommendation diversity while keeping competitive accuracy performance. We have analyzed the influence of different implementation strategies for the IIM module, the tradeoff parameter, the number of interests, as well as different disagreement regularization and maximum entropy regularization on our model's performance. We have also included a case study to show how different modules affect the recommendation results with a concrete example. We found that MDSR_{MA} is superior to MDSR_{DR} in terms of accuracy while MDSR_{DR} outperforms MDSR_{MA} in terms of diversity; the disagreement regularization can both help to improve the performance of MDSR in terms of diversity without much sacrifice in accuracy; and the maximum entropy regularization can help to improve the performance of MDSR in terms of not only accuracy but also diversity. The case study illustrates that MDSR can not only mine users' multiple interests but also produce diverse recommendations to cover and satisfy those interests.

More broadly, our work can be applied to other recommendation tasks for which additional constraints may be applicable, e.g., shared-account recommendations, where the logged behavior may have been produced by multiple users in different periods and thus there multiple interests might be reflected in the sequential behavior [23, 35].

There are various limitations of our work that need to be addressed in the future. First, in MDSR, there is a tradeoff parameter controlling the importance of accuracy vs. diversity, i.e., λ, which should be defined up front. This means that we apply a strategy to provide all users with recommendations in a constant accuracy-diversity balance. However, users may prefer recommendations with different degrees of diversity. For example, in domains where repeat consumption is prevalent, such as retail, diversification needs to be used in a very conservative manner, i.e., without much exploration [42]. However, for domains such as fashion recommendation, the need for exploration is high [32]. It may be helpful for us to incorporate some external information, e.g., category and content information, into our model to adaptively determine the number of interests of each user based his historical interacted items. Second, since our model is based on a sequence-to-list framework, the computational complexity is higher than prior work based on a sequence-to-item framework [34]. It is more reasonable for our model to be applied in the ranking phrase of recommender systems with a small number of candidates, which can help to reduce the computational cost and improve the time efficiency.

As to future work, we want to explore the performance of MDSR when incorporating other effective SR models in our sequence encoder module [47, 54]. Second, it is more reasonable to provide recommendations according to the user's current needs for diversity [13, 57]. Thus, we also plan to explore different ways of learning the tradeoff parameter and making it adaptive to each user. Third, we plan to reduce the computational cost for the training of our model by applying recent efficiency improving techniques [61].

DATA AND CODE

To facilitate the reproducibility of the reported results, this work only made use of publicly available data and our experimental implementation is publicly available at https://bitbucket.org/WanyuChen/idsr/src/master/.

ACKNOWLEDGMENT

We thank the editor-in-chief for her helpful suggestions.

REFERENCES

[1] Adnan Abid, Naveed Hussain, Kamran Abid, Farooq Ahmad, Muhammad Shoaib Farooq, Uzma Farooq, Sher Afzal Khan, Yaser Daanial Khan, Muhammad Azhar Naeem, and Nabeel Sabir. 2016. A survey on search results diversification techniques. *Neural Comput. Appl.* 27, 5 (2016), 1207–1229.

[2] Gediminas Adomavicius and YoungOk Kwon. 2012. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Trans. Knowl. Data Eng.* 24, 5 (2012), 896–911.

[3] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying search results. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*. 5–14.

[4] Azin Ashkan, Branislav Kveton, Shlomo Berkovsky, and Zheng Wen. 2015. Optimal greedy diversity for recommendation. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*. 1742–1748.

[5] Jimmy Ba, Jamie Kiros, and Geoffrey Hinton. 2016. Layer normalization. arXiv:1607.06450.

[6] Sujoy Bag, Abhijeet Ghadge, and Manoj Kumar Tiwari. 2019. An integrated recommender system for improved accuracy and aggregate diversity. *Comput. Industr. Eng.* 130 (2019), 187–197.

[7] Betru Basiliyos, Tilahun, Onana Charles, Awono, and Batchakui Bernabe. 2017. Deep learning methods on recommender system: A survey of state-of-the-art. *Int. J. Comput. Appl.* 162, 10 (2017), 17–22.

[8] Jaime Carbonell and Jade Goldstein. 1998. The Use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 335–336.

[9] Laming Chen, Guoxin Zhang, and Hanning Zhou. 2018. Fast greedy map inference for determinantal point process to improve recommendation diversity. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems*. 5627–5638.

[10] Wanyu Chen, Pengjie Ren, Fei Cai, Fei Sun, and Maarten de Rijke. 2020. Improving end-to-end sequential recommendations with intent-aware diversification. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. 175–184.

[11] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. 108–116.

[12] Peizhe Cheng, Shuaiqiang Wang, Jun Ma, Jiankai Sun, and Hui Xiong. 2017. Learning to recommend accurate and diverse items. In *Proceedings of the 26th International Conference on World Wide Web*. 183–192.

[13] Tommaso Di Noia, Jessica Rosati, Paolo Tomeo, and Eugenio Di Sciascio. 2017. Adaptive multi-attribute diversity for recommender systems. *Inf. Sci.* 382–383 (2017), 234–253.

[14] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AI&Statistics'10)*. 249–256.

[15] John A. Hartigan and M. Anthony Wong. 1979. A k-Means clustering algorithm. *J. Roy. Stat. Soc. Ser. C (Appl. Stat.)* 28, 1 (1979), 100–108.

[16] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 843–852.

[17] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *Proceedings of the 4th International Conference on Learning Representations*. 1–10.

[18] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 241–248.

[19] Geoffrey Hinton, Alex Krizhevsky, and Sida Wang. 2011. Transforming auto-encoders. In *Proceedings of the International Conference on Artificial Neural Networks 2011*. 44–51.

[20] Geoffrey E. Hinton, Sara Sabour, and Nicholas Frosst. 2018. Matrix Capsules with EM routing. In *Proceedings of the Sixth International Conference on Learning Representations.*

[21] Liang Hu, Longbing Cao, Shoujin Wang, Guandong Xu, Jian Cao, and Zhiping Gu. 2017. Diversifying personalized recommendation with user-session context. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. 1858–1864.

[22] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*. 448–456.

[23] Jyun Yu Jiang, Cheng Te Li, Yian Chen, and Wei Wang. 2018. Identifying users behind shared accounts in online streaming services. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. 65–74.

[24] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-attentive sequential recommendation. In *Proceedings of the 18th IEEE International Conference on Data Mining*. 197–206.

[25] Yejin Kim, Kwangseob Kim, Chanyoung Park, and Hwanjo Yu. 2019. Sequential and diverse recommendation with long tail. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2740–2746.

[26] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980. Retrieved from https://arxiv.org/abs/1412.6980.

[27] Matev Kunaver and Toma Porl. 2017. Diversity in recommender systems a survey. *Knowl.-Based Syst.* 123, 1 (2017), 154–162.

[28] 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 *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2615–2623.

[29] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.

[30] Jian Li, Zhaopeng Tu, Baosong Yang, Michael R. Lyu, and Tong Zhang. 2018. Multi-head attention with disagreement regularization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2897—2903.

[31] Shangsong Liang, Emine Yilmaz, Hong Shen, Maarten de Rijke, and W. Bruce Croft. 2017. Search result diversification in short text streams. *ACM Trans. Inf. Syst.* 36, 1 (2017), 8.

[32] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2020. Explainable fashion recommendation with joint outfit matching and comment generation. *IEEE Trans. Knowl. Data Eng.* 32, 8 (Aug. 2020), 1502–1516.

[33] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1831–1839.

[34] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled self-supervision in sequential recommenders. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 483–491.

[35] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. π-Net: A parallel information-sharing network for cross-domain shared-account sequential recommendations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 685–694.

[36] Richard Mccreadie, Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. 2018. Explicit diversification of event aspects for temporal summarization. *ACM Trans. Inf. Syst.* 36, 3 (2018), 25.

[37] Katja Niemann and Martin Wolpers. 2013. A new collaborative filtering approach for increasing the aggregate diversity of recommender systems. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 955–963.

[38] Lijing Qin and Xiaoyan Zhu. 2013. Promoting diversity in recommendation by entropy regularizer. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*. 2698–2704.

[39] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-aware recommender systems. *Comput. Surv.* 51, 4, Article 66 (2018), 36 pages.

[40] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *Proceedings of the 11th ACM Conference on Recommender Systems*. 130–137.

[41] Lakshmanan Rakkappan and Vaibhav Rajan. 2019. Context-aware sequential recommendations with stacked recurrent neural networks. In *Companion Proceedings of the 2019 World Wide Web Conference*. 3172–3178.

[42] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2019. RepeatNet: A repeat aware neural recommendation machine for session-based recommendation. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*. 4806–4813.

[43] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems*. 3856–3866.

[44] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*. 285–295.

[45] Chaofeng Sha, Xiaowei Wu, and Junyu Niu. 2016. A framework for recommending relevant and diverse items. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. 3868–3874.

[46] 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 *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1441–1450.

[47] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N sequential recommendation via convolutional sequence embedding. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. 565–573.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Ł ukasz Gomez, Aidan Nand Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems*. 5998–6008.

[49] Meirui Wang, Pengjie Ren, Lei Mei, Zhumin Chen, Ma Jun, and Maarten de Rijke. 2019. A collaborative session-based recommendation approach with parallel memory modules. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 345–354.

[50] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet Orgun. 2019. Sequential recommender systems: Challenges, progress and prospects. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*. International Joint Conferences on Artificial Intelligence Organization, 6332–6338.

[51] Shoujin Wang, Liang Hu, Yan Wang, Quan Z. Sheng, Mehmet Orgun, and Longbing Cao. 2019. Modeling multi-purpose sessions for next-item recommendations via mixture-channel purpose routing networks. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 3771–3777.

[52] Qiong Wu, Yong Liu, Chunyan Miao, Yin Zhao, Lu Guan, and Haihong Tang. 2019. Recent advances in diversified recommendation. arXiv:1905.06589. Retrieved from https://arxiv.org/abs/1905.06589.

[53] Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip Yu. 2018. Zero-shot user intent detection via capsule neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 3090–3099.

[54] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Jiajie Xu, Victor S.Sheng S.Sheng, Zhiming Cui, Xiaofang Zhou, and Hui Xiong. 2019. Recurrent convolutional neural network for sequential recommendation. In *Companion Proceedings of the 2019 World Wide Web Conference*. 3398–3404.

[55] Min Yang, Wei Zhao, Jianbo Ye, Zeyang Lei, Zhou Zhao, and Soufei Zhang. 2018. Investigating capsule networks with dynamic routing for text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 3110–3119.

[56] Yazhou Yao, Fumin Shen, Jian Zhang, Li Liu, Zhenmin Tang, and Ling Shao. 2019. Extracting privileged information for enhancing classifier learning. *IEEE Trans. Image Process.* 28, 1 (Jan. 2019), 436–450.

[57] Ting Yu, Junpeng Guo, Wenhua Li, Harry Jiannan Wang, and Ling Fan. 2019. Recommendation with diversity: An adaptive trust-aware model. *Decis. Supp. Syst.* 123 (2019), 113073.

[58] Mi Zhang and Neil Hurley. 2008. Avoiding monotony: Improving the diversity of recommendation lists. In *Proceedings of the ACM Conference on Recommender Systems*. 123–130.

[59] Xiao Zhang, Changlin Mei, Degang Chen, and Jinhai Li. 2016. Feature selection in mixed data: A method using a novel fuzzy rough set-based information entropy. *Pattern Recogn.* 56 (2016), 1–15.

[60] Wei Zhao, Haiyun Peng, Steffen Eger, Erik Cambria, and Min Yang. 2019. Towards scalable and reliable capsule networks for challenging NLP applications. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 1549–1559.

[61] Chang Zhou, Jianxin Ma, Jianwei Zhang, Jingren Zhou, and Hongxia Yang. 2020. Contrastive learning for debiased candidate generation in large-scale recommender systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'21)*. 14 pages.

[62] Andrew Zimdars, David Maxwell Chickering, and Christopher Meek. 2001. Using temporal data for making recommendations. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*. 580–588.

Received December 2020; revised April 2021; accepted June 2021