Improving End-to-End Sequential Recommendations with Intent-aware Diversification

Wanyu Chen[1, 2] Pengjie Ren[2] Fei Cai[1, *] Fei Sun[3] Maarten de Rijke[2, 4]
[1] Science and Technology on Information Systems Engineering Laboratory,
National University of Defense Technology, Changsha, China
[2] University of Amsterdam, Amsterdam, The Netherlands
[3] Alibaba Group, Bejing, China
[4] Ahold Delhaize, Zaandam, The Netherlands
{wanyuchen,caifei}@nudt.edu.cn,{p.ren,m.derijke}@uva.nl,ofey.sf@alibaba-inc.com

ABSTRACT

Sequential recommenders that capture users' dynamic intents by modeling sequential behavior, are able to accurately recommend items to users. Previous studies on sequential recommendations (SRs) mostly focus on optimizing the recommendation accuracy, thus ignoring the diversity of recommended items. Many existing methods for improving the diversity of recommended items are not applicable to SRs because they assume that user intents are static and rely on post-processing the list of recommended items to promote diversity. We consider both accuracy and diversity by reformulating SRs as a list generation task, and propose an integrated approach with an end-to-end neural model, called intent-aware diversified sequential recommendation (IDSR). Specifically, we introduce an implicit intent mining (IIM) module for SR to capture multiple user intents reflected in sequences of user behavior. We design an intent-aware diversity promoting (IDP) loss function to supervise the learning of the IIM module and guide the model to take diversity into account during training. Extensive experiments on four datasets show that IDSR significantly outperforms state-of-the-art methods in terms of recommendation diversity while yielding comparable or superior recommendation accuracy.

CCS CONCEPTS

• **Recommender systems**;

KEYWORDS

Diversification; Sequential recommendation

ACM Reference Format:
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 (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340531.3411897

*Corresponding author.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3411897

Figure 1: An example showing sequential recommendations with (bottom) and without (top) diversification.

1 INTRODUCTION

Conventional recommendation methods, e.g., collaborative filtering (CF) based methods [36] or matrix factorization (MF) based models [21], assume that user intents are static. They ignore the dynamic and evolving characteristics of user behavior [28]. Sequential recommenders have been introduced to address these characteristics with the aim of predicting the next item(s) by modeling the sequence of a user's previous behavior [31].

Many early studies on sequential recommendation (SR) are based on Markov chains (MCs) [35], which cannot handle long sequences [15, 16]. Recurrent neural network (RNN) and transformer based neural models have attracted a lot of attention [14, 18] as an alternative. Over the years, many factors have been considered that influence the performance of sequential recommenders, e.g., personalization [32], repeat consumption [34], context [33], and collaboration [41]. Previous work that focuses on these factors usually aims to improve recommendation accuracy only. However, it has been shown that diversity is also an important metric to consider in recommender systems, as users may prefer more diverse lists of recommended items [49].

This is especially true in SR as users may have multiple intents, e.g., different topics or categories of items. For example, as shown in Figure 1, although the user shows most interest in cartoon movies from her historic watching behavior, occasionally she also watches family and action movies. An effective recommendation strategy should provide a diverse list of recommended items so as to satisfy all these intents. Concretely, in the case of Figure 1, we would like to recommend a list of cartoons and action as well as family movies simultaneously instead of cartoons only. Furthermore, user intents are occasionally exploratory, which means that they do not have a specific goal in mind. A homogeneous list of recommendations cannot satisfy such users, which may lead to a boring user experience [37].

Diversification has been well studied in conventional recommendation scenarios [43] as well as in web search [1, 26, 29]. Current approaches to diversified recommendation mainly focus on how to re-rank the items in a list of recommendations based on a given diversity metric with general recommendation models. Such approaches do not constitute an optimal solution for SRs. First, some assume that user intents are static and they require that user intents are prepared beforehand, which is unrealistic in most SR scenarios [7]. Second, most belong to the post-processing paradigm and achieve recommendation accuracy and diversity in two separate steps, i.e., (1) scoring items and generating a candidate item set with a recommendation model; and (2) selecting a diverse list of recommendations based on both the item scores and some implicit/explicit diversity metrics [23, 43]. Because the recommendation models are not aware of diversity during training and it is hard to design optimal diversity strategies for different recommendation models, their performance is unsatisfactory.

In this paper, we address the task of SR by taking into account both recommendation accuracy and diversity. Previous methods focusing on accuracy adopt a strategy where items are ranked by a score, which cannot capture the relationship among the recommended items. Instead, we reformulate SR as a list generation task so as to model the relationship among recommended items and propose an end-to-end intent-aware diversified sequential recommendation (IDSR) model. IDSR employs an *implicit intent mining* (IIM) module to automatically capture multiple latent user intents reflected in sequences of user behavior, and an *intent-aware diversity promoting* (IDP) decoder to directly generate accurate and diverse lists of recommendations for the latent user intents. In order to supervise the learning of the implicit intent mining (IIM) module and force the model to take recommendation diversity into account during training, we design an intent-aware diversity promoting (IDP) loss function that evaluates recommendation accuracy and diversity based on the generated lists of recommended items.

More specifically, a sequence encoder is first used to encode user behavior into representations. Then, the IIM module employs multiple attention areas to mine users' multiple intents with each attention area capturing a particular latent user intent. Finally, an intent-aware recommendation decoder is used to generate a list of recommendations by selecting one item at a time. When selecting the next item, IDSR also takes the items already selected as input so that it can track to what extent each latent user intent is satisfied. During training, we fuse the IDP loss function to learn to mine and track user intents, and recommend diversified items. In order to supervise the learning of diversity, ideally we have a ground truth diverse list of recommended items. However, in practice, we only have the next one ground truth item, which is not enough to define diversity supervision. To address this, we devise a self-critic strategy for the IDP loss. The idea is that, under the premise that the ground truth item can be recommended correctly, we reward our list generation strategy whenever it generates a more diverse recommendation list than the baseline strategy (i.e., the conventional rank-by-score strategy) evaluated by some diversity metrics. All parameters are learned in an end-to-end back-propagation training paradigm within a unified framework.

We conduct extensive experiments on four benchmark datasets. IDSR outperforms the state-of-the-art baselines on those datasets in terms of both accuracy metrics, i.e., Recall and MRR, and a diversity metric, i.e., intra-list distance (ILD).

Our contributions in this paper can be summarized as follows:

- We propose an intent-aware diversified sequential recommendation (IDSR) method. To the best of our knowledge, this is the first end-to-end list generation based neural framework that considers diversification for SRs.
- We devise an implicit intent mining (IIM) module to automatically mine latent user intents from user behavior and an intent-aware recommendation decoder to generate diverse lists of recommendations.
- We present an IDP loss function to supervise IDSR in terms of both accuracy and diversity.
- We carry out extensive experiments and analyses on four publicly available benchmark datasets to verify the effectiveness of the proposed IDSR.

2 RELATED WORK

We discuss two types of work that is closely related to ours: sequential recommendation and diversified recommendation.

2.1 Sequential recommendation

Traditional methods for SRs are often based on Markov chains (MCs) [51]. Previous work introducing such methods investigates how to extract sequential patterns to learn users' next preferences with probabilistic decision-tree models. Following this idea, He and McAuley [12] fuse similarity models with MCs to address the problem of sparse recommendations. MC-based methods only model local sequential patterns with adjacent interactions, which fails to take the whole sequence into account.

Hidasi et al. [15] introduce an RNN-based model for SRs that consists of gated recurrent units (GRUs) and uses a session-parallel mini-batch training process. Quadrana et al. [32] develop a hierarchical RNN structure that takes users' profiles into account by considering cross-session information. Attention mechanisms have been applied to SRs to help models explore users' preferences [13]. Li et al. [24] propose a neural attentive session-based recommendation machine that takes the last hidden state from a session-based RNN as the sequential behavior, and uses the other hidden states for computing attention to capture users' current preferences in a given session. Xu et al. [44] propose a recurrent convolutional neural network to capture both long-term and short-term dependencies for SR. Kang and McAuley [18] apply a two-layer transformer model [40] to SRs to capture users' sequential behavior. Sun et al. [38] use a bidirectional encoder representations from transformers for SRs. Chen et al. [8] propose to apply a user memory network with attention mechanism to store and update a user's historical records for SRs.

Previous studies on SRs, e.g., [4, 6, 46, 47], mostly focus on improving the recommendation accuracy. The studies mentioned above ignore the fact that users might have multiple intents reflected in their sequential behavior. Wang et al. [42] have proposed a mixture-channel purpose routing networks (MCPRNs) to capture users' different intents in a given session. MCPRN first applies a purpose routing network to detect multiple purposes of a user and then models the items with a mixture-channel RNN, where each

channel RNN models the item dependencies for a specific purpose. Finally, MCPRN integrates all channel embeddings to predict the next item. During training, MCPRN only applies the cross-entropy loss to supervise the model in terms of recommendation accuracy, which means there is no supervision for the model to learn to distinguish multiple intents or generate diversified recommendations.

Unlike the studies listed above, we propose to address recommendation accuracy and diversification in a unified framework, where we propose an implicit intent mining (IIM) module to capture multiple intents and an intent-aware diversity promoting (IDP) decoder to generate the list of recommended items to satisfy those intents gradually. We devise an IDP loss function to supervise the model to learn different intents and generate diversified recommendations.

2.2 Diversified recommendation

Promoting diversity of recommendation or search results has long been an important research topic. A lot of work has proposed to tackle the task of diversified recommendation, mainly including determinantal point process (DPP) [22] and submodular optimization [30]. The most representative implicit approach is maximal marginal relevance (MMR) [5]. MMR represents relevance and diversity by independent metrics and uses the notion of marginal relevance to combine the two metrics with a trade-off parameter. Qin and Zhu [30] propose an entropy regularizer to promote recommendation diversity. It satisfies monotonicity and submodularity so that the objective function can be maximized approximately by a greedy algorithm. Chen et al. [7] propose to improve recommendation diversification through a DPP [22] with a greedy maximum a posterior inference algorithm. Sha et al. [37] introduce a submodular objective function to combine relevance, coverage of user's intents, and the diversity between items. Learning to rank (LTR) has also been exploited to address diversification. Cheng et al. [9] first label each user by a set of diverse as well as relevant items with a heuristic method and then propose a diversified collaborative filtering algorithm to learn to optimize the performance of accuracy and diversity for recommendation. The main issue of LTR based methods is that they all need diversified ranked lists as ground truth for learning [43]; these are usually unavailable in recommendations.

The methods listed above achieve accuracy and diversity of recommendation in two separate steps, i.e., training an offline recommendation model to score items in terms of accuracy and then re-ranking items by taking diversity into account. We show through experiments that our end-to-end model can achieve significantly better performance. Besides, none of the methods listed is suitable for SRs, where users' sequential behavior needs to be considered. In contrast, we consider users' temporal preferences and optimize for accuracy and diversity in one go.

3 INTENT-AWARE DIVERSIFIED SEQUENTIAL RECOMMENDATIONS

3.1 Overview

Given a user u and her/his behavior sequence $S_u = \{x_1, x_2, \ldots, x_T\}$ where every x_i is an item that u interacted with, e.g., watched movie, the goal of SRs is to provide u with a list of recommended items R_L for predicting her/his next interaction; the items are expected to be both relevant and diverse.

Unlike existing SR methods, we assume there are M latent intents behind each behavior sequence, i.e., $A = \{a_1, \ldots, a_M\}$. Then, we seek to generate a list of recommended items R_L by maximizing the degree of satisfaction for all intents:

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

where $P(a_m \mid u)$ denotes the importance of intent a_m to user u; $P(R_L \mid a_m, u, S_u)$ is the probability of satisfaction of R_L to a_m.

It is hard to directly optimize $P(R_L \mid u, S_u)$ due to the huge search space. Therefore, we propose to generate R_L greedily, i.e., selecting one item at a time with the maximum score $S(v)$:

$$v_t \leftarrow \underset{v \in V \setminus R_{t-1}}{\arg\max}\ S(v), \tag{2}$$

where v_t is the item to be selected at step t; V is the set of all items; R_{t-1} is the list of recommended items generated until step t-1; $V \setminus R_{t-1}$ guarantees that the selected item is different from previous generated recommendations in R_{t-1} at step t; and $S(v)$ returns the score of item v by

$$S(v) \leftarrow \lambda P(v \mid u, S_u) + (1-\lambda) \sum_{m=1}^{M} P(v \mid a_m) W(\overline{R_{t-1}}, a_m). \tag{3}$$

The score $S(v)$ is a combination of the relevance score and the diversification score, balanced by a hyper-parameter λ; $P(v \mid u, S_u)$ is the relevance score reflecting the importance of v for u; $P(v \mid a_m)$ is the degree of satisfaction of v to a_m; $W(\overline{R_{t-1}}, a_m)$ denotes the likelihood that the already generated recommendation list R_{t-1} does not satisfy a_m.

Then, we propose an end-to-end intent-aware diversified sequential recommendation (IDSR) model to directly generate a diversified list of recommended items according to Eq. (3). The main framework of IDSR is shown in Figure 2. As shown in Figure 2, IDSR consists of three modules: a *sequence encoder*, an *implicit intent mining* (IIM) *module*, and an *intent-aware diversity promoting* (IDP) *decoder*. First, the sequence encoder projects users' sequential behavior into latent representations. Then, the IIM module captures users' multiple latent intents reflected in their sequential behavior. Finally, the IDP decoder is employed to generate a list of recommended items according to Eq. (3). We devise an IDP loss to train IDSR; it evaluates the whole list of recommended items in terms of both accuracy and diversity. Note that there is no re-ranking involved in IDSR. Recommendation accuracy and diversity are jointly learned in an end-to-end way. Next, we introduce the separate modules.

3.2 Sequence encoder

Since the encoder module is not the focus of this paper, we simply adapt the commonly used GRUs to verify the validity of our proposed method [15]:

$$\begin{aligned} z_t &= \sigma\big(\boldsymbol{W}_z[\mathrm{x}_t, \boldsymbol{h}_{t-1}]\big) \\ r_t &= \sigma\big(\boldsymbol{W}_r[\boldsymbol{x}_t, \boldsymbol{h}_{t-1}]\big) \\ \hat{\boldsymbol{h}}_t &= \tanh(\boldsymbol{W}_h[\boldsymbol{x}_t, r_t \odot \boldsymbol{h}_{t-1}]) \\ \boldsymbol{h}_t &= (1-z_t) \odot \boldsymbol{h}_{t-1} + z_t \odot \hat{\boldsymbol{h}}_t, \end{aligned} \tag{4}$$

where $\boldsymbol{x}_t$ denotes the embedding of item x_t; $\boldsymbol{W}_z$, $\boldsymbol{W}_r$ and $\boldsymbol{W}_h$ are weight parameters; σ denotes the sigmoid function. The input of

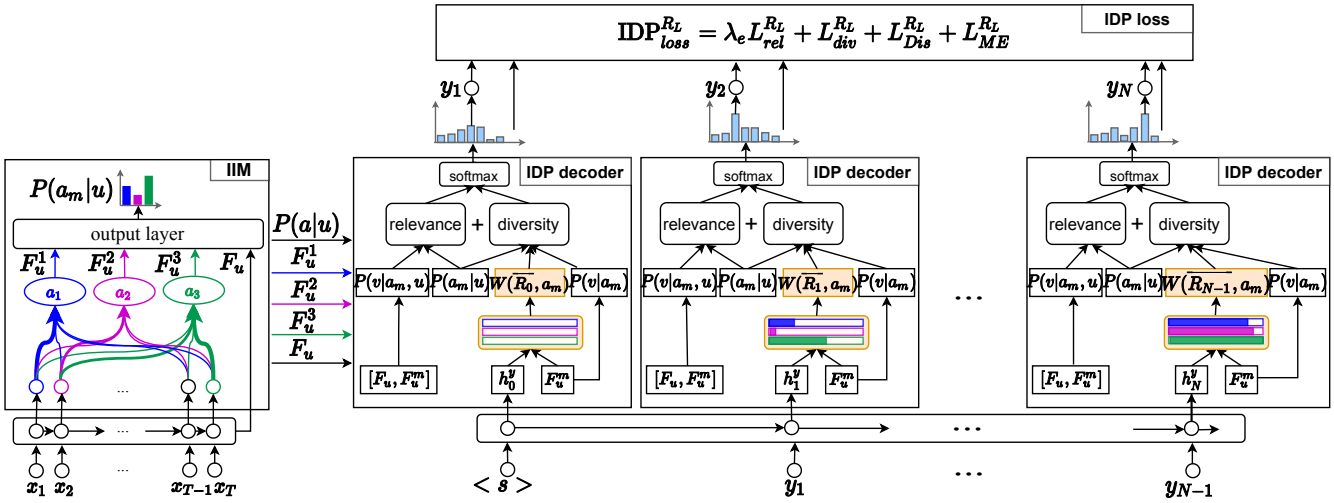


Figure 2: Overview of IDSR. The blue, purple and green colors denote different user intents.

the encoder is the behavior sequence $S_u = \{x_1, x_2, \ldots, x_T\}$ and the outputs are hidden representations $\{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_T\}$, where $\boldsymbol{h}_i \in \mathbb{R}^{d_e}$. We stack those representations into a matrix $\boldsymbol{H}_S \in \mathbb{R}^{T \times d_e}$. Like [24], we consider the last representation $\boldsymbol{h}_T$ to be the user's global representation, which summarizes the whole sequence:

$$F_u = \boldsymbol{h}_T. \tag{5}$$

3.3 IIM module

The IIM module is meant to mine users' multiple intents behind the sequence. Intuitively, a user's multiple intents can be reflected by different interactions in their sequential behavior. Some interactions are more representative for a particular intent than others, e.g., the last two actions in Figure 1 reflect the user's intent of watching cartoon movies. Motivated by this, we fuse a multi-intent attention mechanism where each attention function captures one particular intent. Specifically, IIM first projects $\boldsymbol{H}_S$ and F_u into M spaces w.r.t. the latent intents, respectively. Then, M attention functions are employed in parallel to produce user's intent-specific representations $\{S_u^1, S_u^2, \ldots, S_u^M\}$:

$$S_u^i = \text{Attention}(F_u \boldsymbol{W}_i^Q, \boldsymbol{H}_S \boldsymbol{W}_i^K, \boldsymbol{H}_S \boldsymbol{W}_i^V), \tag{6}$$

where the projection matrices for intent i, i.e., $\boldsymbol{W}_i^Q \in \mathbb{R}^{d_e \times d}$, $\boldsymbol{W}_i^K \in \mathbb{R}^{d_e \times d}$ and $\boldsymbol{W}_i^V \in \mathbb{R}^{d_e \times d}$, are learnable parameters. We use the scaled dot-product attention in this work [40] as:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \boldsymbol{A}\boldsymbol{V} = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d}}\right)\boldsymbol{V}, \tag{7}$$

where $\boldsymbol{A}$ denotes the attention distribution produced by each intent. We finally apply a two-layer feed-forward network to each S_u^i to introduce nonlinearity:

$$F_u^i = \text{FFN}(S_u^i) = \text{ReLU}(S_u^i \boldsymbol{W}^{(1)} + \boldsymbol{b}^{(1)})\boldsymbol{W}^{(2)} + \boldsymbol{b}^{(2)}, \tag{8}$$

where $\boldsymbol{W}^{(1)} \in \mathbb{R}^{d \times d}$, $\boldsymbol{W}^{(2)} \in \mathbb{R}^{d \times d}$, $\boldsymbol{b}^{(1)} \in \mathbb{R}^d$, and $\boldsymbol{b}^{(2)} \in \mathbb{R}^d$ are trainable parameters.

3.4 IDP decoder

The IDP decoder is used to generate R_L based on the intents mined with the IIM module. To begin, we model the relevance score of v to user u (i.e., $P(v \mid u, S_u)$ in Eq. (3)) with a bilinear decoding scheme as follows:

$$\begin{aligned} P(v_n \mid u, S_u) &= \frac{S_{\boldsymbol{v}_n}}{\sum_{j=1}^{|V|} S_{\boldsymbol{v}_j}} \\ S_{v_n} &= \sum_{m=1}^{M} S_{v_n}^m \\ S_{v_n}^m &= P(a_m \mid u) P(v_n \mid a_m, u) \\ P(v_n \mid a_m, u) &= \text{softmax}(\boldsymbol{v}_n^\top \boldsymbol{B} [F_u, F_u^m]), \end{aligned} \tag{9}$$

where $\boldsymbol{B}$ is a bilinear parameter; $\boldsymbol{v}_n$ is the item embedding which can be trained within the network; and $S_{v_n}^m$ means the relevance score of item v_n to intent a_m, weighted by the importance of intent a_m, i.e., $P(a_m \mid u)$. We can calculate $P(a_m \mid u)$ by:

$$P(a_m \mid u) = \frac{\exp(F_u \boldsymbol{W}^w F_u^{m\top})}{\sum_{j=1}^{M} \exp(F_u \boldsymbol{W}^w F_u^{j\top})}, \tag{10}$$

where $\boldsymbol{W}^w \in \mathbb{R}^{d_e \times d}$ is used to transform the intent-specific representations back to the same space with F_u, so that we can generate the weight of each intent.

To track the already selected items to date, we use another GRU to encode $R_{t-1} = \{y_1, y_2, \ldots, y_{t-1}\}$ into $\{\boldsymbol{h}_1^y, \boldsymbol{h}_2^y, \ldots, \boldsymbol{h}_{t-1}^y\}$. Then we estimate the degree of "unsatisfactoriness" of R_{t-1} to each intent (i.e., $W(\overline{R_{t-1}}, a_m)$ in Eq. (3)) by calculating the matching between $\boldsymbol{h}_{t-1}^y$ and F_u^m as:

$$\begin{aligned} W(\overline{R_{t-1}}, a_m) &= 1 - \frac{P(a_m \mid u) \exp(w_{t-1}^m)}{\sum_{j=1}^{M} P(a_j \mid u) \exp(w_{t-1}^j)} \\ w_{t-1}^i &= \boldsymbol{W}_{\boldsymbol{y}}^\top \sigma(\boldsymbol{W}_{\boldsymbol{A}} F_u^i + \boldsymbol{W}_{\boldsymbol{B}} \boldsymbol{h}_{t-1}^y), \end{aligned} \tag{11}$$

where w_{t-1}^i denotes the matching between already generated recommendations and F_u^i. Thus $W(\overline{R_{t-1}}, a_m)$ indicates to what extent intent a_m is unsatisfied and should be paid more attention to when generating the next recommendation. Here, we also incorporate the initial weight of each intent $P(a \mid u)$. We calculate $P(v \mid a_m)$ in Eq. (3) with:

$$P(v_n \mid a_m) = \text{softmax}(\boldsymbol{v}_n^\top F_u^m). \tag{12}$$

Finally, we can calculate the score $S(v)$ of each item (Eq. (3)), select the item with the highest probability, and append it to the list of recommended items.

3.5 IDP loss

Since our goal is to generate a list of recommended items that is both relevant and diverse, we design our loss function to evaluate the whole generated list R_L based on the accuracy as well as the diversity of R_L:

$$\text{Loss}^{R_L} = \lambda_e \mathcal{L}_{rel}^{R_L} + \mathcal{L}_{div}^{R_L}, \tag{13}$$

where λ_e is a weight parameter to balance the relative contributions of accuracy and diversification.

Given the output list of recommended items from IDSR, i.e., $R_L = \{y_1, y_2, \ldots, y_N\}$ and the ground truth item y^* (i.e., the next consumed item), $\mathcal{L}_{rel}^{R_L}$ is defined as:

$$\mathcal{L}_{rel}^{R_L} = -\sum_{i=1}^{|V|} p_i \log(q_i^0), \tag{14}$$

where p_i indicates the ground truth probability distribution and q_i^0 is the prediction probability of the first item in R_L. When generating the first item, IDSR only considers the relevance score without diversification, thus we use this part to optimize the prediction accuracy of IDSR. With this relevance loss, we can also take the position of the ground truth item y^* in the ranked list into consideration.

To promote diversity, we apply a self-critic strategy. Specifically, at each step, we select an item based on $S(v)$ and output a list of recommended items R_L. Meanwhile, we also select an item only based on the maximum relevance score $P(v_i \mid u, S_u)$ and output a list of recommended items R_L^{rel}. Thus we propose a pair-wise diversity loss:

$$\begin{aligned} \mathcal{L}_{div}^{R_L} &= \boldsymbol{w} \log \frac{1}{1 + \exp(Pr(R_L^{rel}) - Pr(R_L))} \\ Pr(R_L) &= \sum_{v_i \in R_L} \log S(v_i) \\ Pr(R_L^{rel}) &= \sum_{v_i \in R_L^{rel}} \log S(v_i) \\ \boldsymbol{w} &= \boldsymbol{M}(R_L^{rel}) - \boldsymbol{M}(R_L), \end{aligned} \tag{15}$$

where $S(v_i)$ is the final score of item v_i calculated by Eq. (3); $Pr(R_L)$ indicates the log likelihood of generating recommendation list R_L, so as R_L^{rel}; $\boldsymbol{w}$ is the diversity evaluation metric score gap of the two recommendation list R_L^{rel} and R_L, e.g., ILD in this paper. We use R_L^{rel} as a baseline to compare with, so that we can evaluate the diversity of the generated list of recommended items R_L. If the diversity of R_L^{rel} is larger than R_L, we would punish the decoder to decrease the probability for generating R_L with the weight of $\boldsymbol{w}$. Otherwise, we would reward the decoder to increase probability of R_L, which is larger than the probability of generating R_L^{rel}.

Besides the relevance and diversity losses, we also add two regularization terms to our loss function. One is a disagreement regularization, which is meant to enlarge the distance among multiple intents. Specifically, the differences among multiple intent representations are reflected by different attention distributions produced by each intent, thus we apply a strategy to disperse the attended positions predicted by each intent. We use an alignment disagreement regularization [25] as:

$$\mathcal{L}_{Dis}^{R_L} = \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \left| \boldsymbol{A}^i \odot \boldsymbol{A}^j \right|, \tag{16}$$

where $\boldsymbol{A}^i$ denotes the attention distribution produced by intent i in Eq. (7). We employ the sum of element-wise multiplication of vector cells.

The other regularization term that we add is maximum entropy regularization, which helps to avoid the situation that one of the intents dominates [45, 50]:

$$\mathcal{L}_{ME}^{R_L} = \sum_{m=1}^{M} P(a_m \mid u) \log P(a_m \mid u). \tag{17}$$

Thus, our final IDP loss is:

$$\text{IDP}_{loss}^{R_L} = \lambda_e \mathcal{L}_{rel}^{R_L} + \mathcal{L}_{div}^{R_L} + \mathcal{L}_{Dis}^{R_L} + \mathcal{L}_{ME}^{R_L}. \tag{18}$$

All parameters of IDSR as well as the item embeddings can be learned in an end-to-end back-propagation training paradigm.

4 EXPERIMENTAL SETUP

We design experiments to answer the following research questions:

(**RQ1**) What is the performance of IDSR compared with state-of-the-art baselines in terms of accuracy?

(**RQ2**) Does IDSR outperform state-of-the-art baselines in terms of diversity?

4.1 Datasets

We use four public benchmark datasets for our experiments, two of them are based on movies and the others are e-commerce datasets. Table 1 lists the statistics of these four datasets:

- **ML100K**[1] is collected from the MovieLens web site. It contains 100,000 ratings from 943 users on 1,682 movies.
- **ML1M**[1] is a larger and sparser version of ML100K, which contains 1,000,209 ratings for movies.
- **Tafeng**[2] is collected from a grocery store and released by Kaggle, which contains one month log data.
- **Tmall**[3] is released by a competition that records user online shopping behavior on an e-commerce platform, Tmall.

Note that each item/movie from both ML1M and ML100K belongs to multiple movie genres at the same time. Each item from Tafeng and Tmall only belongs to a single category.

Table 1: 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

[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

We follow Li et al. [24] to process the data. First, we filter out users who have less than 5 interactions and items that are rated less than 5 times in ML100K. For the other datasets, we only keep users as well as items with more than 20 interactions. Then, we sort the interactions according to the "timestamp" field to get a behavioral sequence for each user. Finally, we prepare each data sample using a sliding-window approach by regarding the previous 9 actions as input and the next action as output. We use the first 90% interactions for model training, the last 10% for model testing. The validation set is split from the training set in the same way as the test set.

Since we do not target cold-start items, we make sure that all items in the test set have been rated by at least one user in the training set and the test set contains the most recent actions which happened later than those in the training and validation sets.

4.2 Methods used for comparison

A number of SR methods have been proposed in recent years. In our modeling we focus on combining recommendation accuracy and diversity in a unified framework. Thus we do not make comparisons with work that is exclusively aimed at improving recommendation accuracy, e.g., BERT4Rec [38] and SASRec [18], as such work can be incorporated into our encoder to help improve accuracy. There is previous work, i.e., S-DIV [19], that proposes a sequential and diverse recommendation model. But since in S-DIV the term "diverse" means to incorporate more rare or tail items which is different from our work, we do not compare with it in this paper. For a fair comparison, we select state-of-the-art neural SR methods that use a similar architecture as ours as baselines.

- **GRU4Rec**: An RNN-based model for SR. GRU4Rec utilizes session-parallel mini-batches as well as a ranking-based loss function in the training process [15].
- **NARM**: An RNN-based model that applies an attention mechanism to capture users' main purposes from the hidden states and combines it with sequential behavior as final representations of users' current preferences [24], which shares a similar spirits as IDSR when calculating the relevance scores for items.
- **MCPRN**: The most recently proposed method that models users' multiple purposes in a session [42]. The authors claim that they can improve the performance over the state-of-the-art methods in terms of both accuracy and diversity. Thus, we consider it as a state-of-the-art baseline model.

We also report results of a popularity based method, **POP**, which ranks items based on the number of interactions, because the performance of **POP** can reflect characteristics of the datasets and is quite effective in some scenarios [2].

Because there is no previous work specific for diversified SR, we construct a baseline, **NARM+MMR**, ourselves. With carefully tuned hyperparameters, NARM can achieve state-of-the-art performance most of the time. MMR is a simple yet effective approach, which is still commonly used in web search and recommendation. Specifically, we first get the relevance scores $S(v)$ for each item with NARM. Then, we rerank the items using the MMR criteria:

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

where R_c is a candidate item set and $\theta \in [0,1]$ is a trade-off parameter to balance the relevance and the minimal dissimilarity d_{ki} between item v_k and item v_i. MMR first initializes $R_L = \emptyset$ and then iteratively selects the item into R_L, until $|R_L| = N$. When $\theta = 0$, MMR returns diversified recommendations without considering relevance; when $\theta = 1$, it returns the same results as the original baseline models. Unless specified otherwise, for all the results that we presented in the paper, the number of recommendations (N) equals 10.

4.3 Evaluation metrics

To evaluate accuracy, we use Recall and MRR as most previous studies [24, 27]; to evaluate diversity, we choose ILD [49], which is commonly used to evaluate the recommendation diversity.

- **Recall**: Measures whether the test item is contained in the list of recommendations.
- **MRR**: Measures whether the test item is ranked at the top of the list.
- **ILD**: Measures the diversity of a list of recommendations as the average distance between pairs of recommended items:

$$\text{ILD} = \frac{2}{|R_L|(|R_L|-1)} \sum_{(i,j) \in R_L} d_{ij}. \tag{19}$$

We calculate the dissimilarity d_{ij} between two items based on the Euclidean distance between the item genre vectors [3].

4.4 Implementation details

We set the item embedding size and GRU hidden state sizes to 128. We use dropout with drop ratio $p = 0.5$. We initialize the model parameters randomly using the Xavier method [11]. We optimize the model using Adam [20], with the initial learning rate $\alpha = 0.001$, two momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The mini-batch size is set to 512. We set the parameter $\lambda_e = 1.0$ for the ML100K, ML1M and Tmall datasets and $\lambda_e = 0.1$ for Tafeng after fine-tuning the parameter on the validation set. We test the model performance on the validation sets for every epoch and select the best model to report results on the test sets accordingly. The code used to run our experiments is available online.

5 RESULTS

5.1 Performance in terms of accuracy

To answer RQ1, we compare IDSR with the baselines in terms of Recall and MRR; see Table 2.

First, note that the neural attentive recommendation machine (NARM) has a similar encoding architecture as IDSR, thus we can see that NARM and IDSR are comparable in terms of recommendation accuracy (Recall and MRR). However, IDSR can help to improve the diversity (see Section 5.2) of the list of recommendations without much sacrifice of accuracy, i.e., a 0.87% and 1.70% decrease in terms of Recall and MRR on the ML1M dataset, and of 0.65% and 1.01% on the Tmall dataset, respectively, none of which are significant. Although IDSR tries to diversify the recommendations, it still assigns high probability to the most relevant items without considering much of the diversification in the first few decoding steps. In addition, the IDP loss also considers recommendation accuracy, which can help the model to capture users' main intents. When users have multiple intents, NARM shows bias towards the main intent, which will lead to unsatisfactory recommendations.

Table 2: Performance of recommendation models. The results from the best baseline and the best performer in each column are underlined and boldfaced, respectively. Statistical significance of pairwise differences of IDSR vs. the best baseline is determined by a paired t-test ($^{\Delta}$ for p-value $\leq$ .05).

Model	ML100K			ML1M			Tafeng			Tmall		
	Recall (%)	MRR (%)	ILD	Recall (%)	MRR (%)	ILD	Recall (%)	MRR (%)	ILD	Recall (%)	MRR (%)	ILD
POP	4.02	1.21	1.501	9.11	2.02	1.233	2.01	1.09	1.233	9.56	4.11	.8817
GRU4Rec	6.23	2.09	1.527	11.67	4.02	1.307	4.11	1.42	1.267	12.11	5.41	.8789
NARM	9.68	3.18	1.518	**15.02**	**5.39**	1.289	4.71	1.69	1.214	**14.41**	**7.51**	.8343
MCPRN	9.27	2.99	1.561	14.89	5.26	1.301	4.57	1.60	1.248	14.19	7.27	.8864
NARM+MMR	9.53	2.77	1.583	14.72	4.89	1.325	4.33	1.41	1.263	14.00	6.28	.8917
IDSR	**9.79**	**3.22**	**1.666**$^{\Delta}$	14.89	5.30	**1.383**$^{\Delta}$	**4.97**$^{\Delta}$	**1.96**$^{\Delta}$	**1.318**$^{\Delta}$	14.32	7.43	**.9468**$^{\Delta}$

For example, IDSR shows better performance than NARM on the Tafeng dataset. The improvements of IDSR over NARM in terms of Recall and MRR are 5.61% and 16.23% on the Tafeng dataset, respectively. We believe that this is due to the fact that Tafeng records users' behavior in a grocery store, where users tend to have multiple intents, and buy items with different categories when they are shopping. Compared with MCPRN, we can see that IDSR shows better performance in terms of both Recall and MRR on all datasets than MCPRN. The IIM module considers not only users' multiple intents but also the importance of each intent, which can help to improve the recommendation accuracy.

Second, we note that after re-ranking with MMR, the accuracy of NARM drops dramatically, especially in terms of MRR. This indicates that although post-processing with MMR can improve the diversity of recommendation list, it hurts the accuracy a lot. Because most of the candidate items generated by NARM have similar genres/characteristics. When the diversity scores for the relevant items are lower than the irrelevant ones, the irrelevant items will get higher final scores than the relevant items, which results in a worse performance in terms of accuracy. Besides, we found that the re-ranking process is time-consuming, while our model is much more efficient.

In summary, IDSR can achieve comparable or superior performance compared with state-of-the-art methods in terms of recommendation accuracy. It is also worth noting that we can incorporate any other effective mechanisms into our framework to further improve the recommendation accuracy such as SASRec [18]. However, this is beyond the scope of this work.

5.2 Performance in terms of diversity

To answer RQ2, we report the diversity scores, i.e., ILD, on all datasets in Table 2. We can see that IDSR consistently outperforms all baselines. The improvements of IDSR over MCPRN are 6.71% and 6.33% in terms of ILD on ML100K and ML1M, respectively. As to the e-commerce datasets, the improvements are 5.58% and 6.81% on the Tafeng and Tmall datasets, respectively. Although MCPRN models users' multiple intents, there is no supervision signal for the model to learn to distinguish different intents in order to generate diverse recommendations. However, in IDSR, we have the diversity loss and disagreement regularization term in the IDP loss, which helps the model to learn to distinguish different intents and satisfy each of them during the recommendation list generation process.

Table 3: Performance of IDSR with different number of intents.

Dataset	Metric	1-head	2-head	3-head	4-head
ML100K	Recall (%)	9.99	9.83	9.79	9.41
	MRR (%)	3.29	3.19	3.22	2.99
	ILD	1.57	1.62	1.67	1.67
ML1M	Recall (%)	15.26	14.93	14.89	14.01
	MRR (%)	5.55	5.36	5.30	5.02
	ILD	1.29	1.29	1.38	1.40
Tafeng	Recall (%)	5.35	5.16	4.97	4.97
	MRR (%)	1.79	2.02	1.96	1.84
	ILD	1.26	1.26	1.32	1.34
Tmall	Recall (%)	14.51	14.36	14.32	14.21
	MRR (%)	7.49	7.38	7.43	7.23
	ILD	0.82	0.90	0.95	0.95

Clearly, IDSR significantly outperforms NARM+MMR. For example, the improvements of IDSR over NARM+MMR are 4.34% and 6.18% on Tafeng and Tmall, respectively. Since MMR is heuristically defined, we find that MMR relies heavily on the performance of NARM. When the candidate items from NARM all have similar genres, the performance of MMR method is limited. In contrast, IDSR avoids this issue by learning to diversify the recommendation list through optimizing the IDP loss in Eq. (18).

6 ANALYSIS

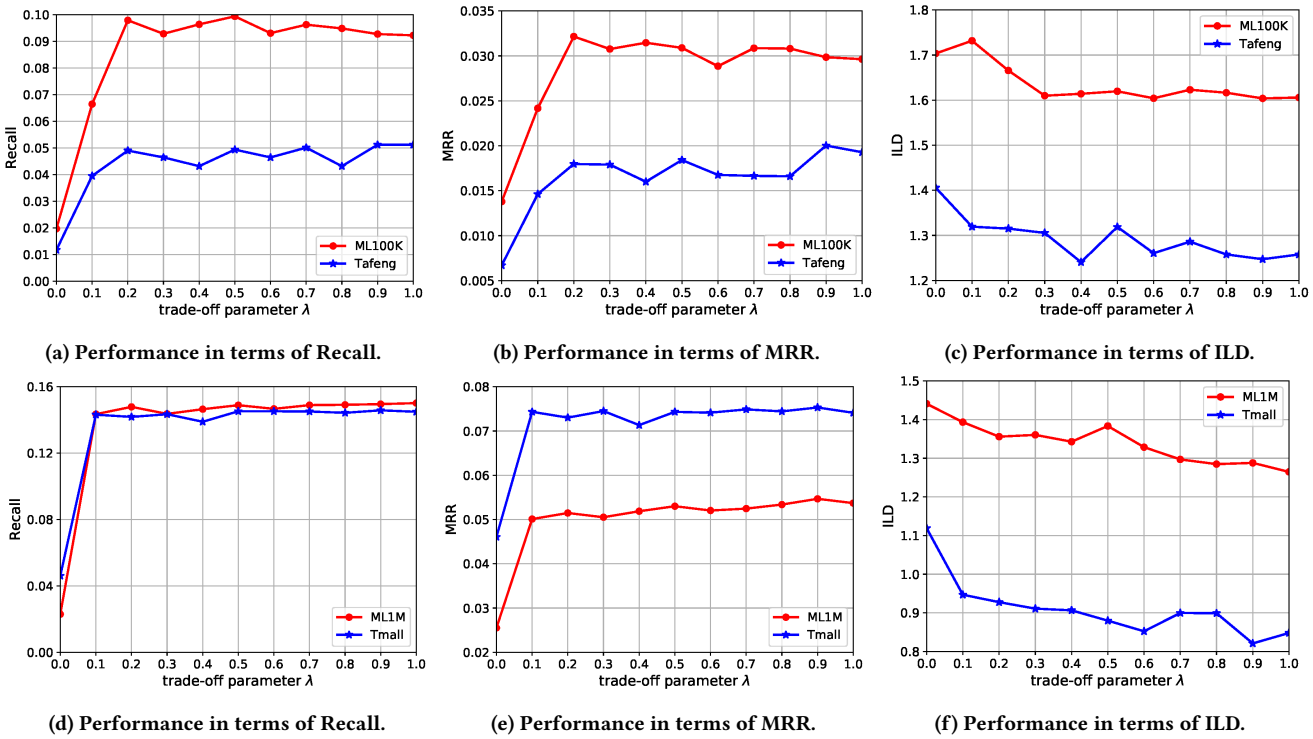
In this section, we perform a number of analyses of the factors that impact the performance of IDSR:

- What is the impact of the number of latent intents on IDSR, i.e, IDSR with a single head or multiple heads?
- How does the trade-off parameter λ affect the performance of IDSR?
- What is the effect of the disagreement regularization loss $\mathcal{L}_{Dis}^{R_L}$ in Eq. (18)?
- Does the IIM module in IDSR capture users' multiple intents?

6.1 Impact of the number of latent intents

We examine the performance of IDSR with different numbers of latent intents/attention heads in Table 3.

We can see that when the number of heads is set to one, the performance is inferior in terms of diversity on all datasets. The



(a) Performance in terms of Recall. (b) Performance in terms of MRR. (c) Performance in terms of ILD.

(d) Performance in terms of Recall. (e) Performance in terms of MRR. (f) Performance in terms of ILD.

Figure 3: Performance of IDSR on four datasets with the parameter λ in Eq. (3) changing from 0 to 1.

reason is that the model will only focus on the main intent when generating recommendations.

As for accuracy, we can see that with the number of heads increasing, the performance in terms of MRR and Recall is getting worse in general. On the e-commerce dataset, i.e., Tmall, the differences in terms of Recall and MRR when we change our model from single head to multiple heads are smaller than those on the MovieLens dataset, e.g., ML1M. The improvement of IDSR with 4-heads over 1-head in terms of ILD on Tmall is larger than that on ML1M. This may be because users are more likely to have multiple intents when they do online shopping than when choosing movies to watch next. Another reason is that the time gap between adjacent interactions in the MovieLens datasets is larger than that in the e-commerce datasets, so historical behavior and multiple intents do not have much impact on users' current behavior.

Table 3 shows that adding more heads will hurt the accuracy much and also increases the number of parameters for training, thus we choose to use three heads in our experiments which are tuned on the validation set.

6.2 Influence of the trade-off parameter λ

In order to investigate the impact of the trade-off parameter λ on IDSR, we test the performance of IDSR on all datasets by ranging λ from 0 to 1 with a step size of 0.1. The results are shown in Figure 3.

The accuracy metrics, i.e., Recall and MRR, show upward trends when λ increases from 0 to 1. When $\lambda = 0$, IDSR shows the worst performance. However, a noticeable increase is observed when λ changes from 0 to 0.1: the setting with $\lambda = 0$ means that we only consider diversity without accuracy, thus the model cannot be trained well to recommend relevant items. IDSR shows its best performance in terms of accuracy metrics with λ at around 0.2 and 0.5 on the ML100K and ML1M datasets. Similar trends can be found on e-commerce datasets in terms of MRR and Recall.

Regarding recommendation diversity, IDSR achieves the best performance in terms of ILD when $\lambda = 0.0$ on all datasets since we maximize diversity only in this case. When λ changes from 0 to 1, ILD naturally decreases on all datasets. On the e-commerce datasets, there are more fluctuations than on the MovieLens datasets, especially on Tmall. The performance of IDSR in terms of ILD decreases sharply from 0 to 0.1.

6.3 Effect of disagreement regularization

In order to look into the effect of the disagreement regularization loss $\mathcal{L}_{Dis}^{R_L}$ is IDSR, we modify the IDP loss as:

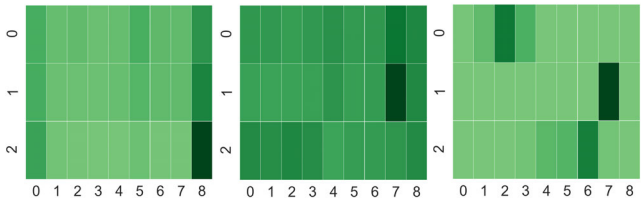
$$\text{IDP}_{loss}^{R_L} = \lambda_e \mathcal{L}_{rel}^{R_L} + \mathcal{L}_{div}^{R_L} + \lambda_{Dis} \mathcal{L}_{Dis}^{R_L} + \mathcal{L}_{ME}^{R_L}, \tag{20}$$

where $\mathcal{L}_{Dis}^{R_L}$ is weighted by the parameter λ_{Dis}. We test the performance of IDSR with $\lambda_{Dis} = 0.0$, 0.5 and 1.0, respectively. The results are shown in Table 4.

We can see that $\mathcal{L}_{Dis}^{R_L}$ can help to boost the performance of IDSR in terms of diversity when λ_{Dis} changes from 0.0 to 1.0. This indicates that the IIM module can effectively capture different latent intents by applying $\mathcal{L}_{Dis}^{R_L}$. To further show the effect of the IIM module with different weights of $\mathcal{L}_{Dis}^{R_L}$, we randomly select one sequence from the test set of ML100K and visualize the attention weights of different positions with multiple intents when $\lambda_{Dis} = 0.0$, 0.5 and 1.0 in Figure 4.

Table 4: Performance of IDSR with different weights of disagreement regularization.

Dataset	Metric	$\lambda_{Dis} = 0.0$	$\lambda_{Dis} = 0.5$	$\lambda_{Dis} = 1.0$
ML100K	Recall (%)	9.95	9.78	9.79
	MRR (%)	3.23	3.13	3.21
	ILD	1.58	1.61	1.67
ML1M	Recall (%)	15.14	14.97	14.89
	MRR (%)	5.48	5.32	5.30
	ILD	1.29	1.30	1.38
Tafeng	Recall (%)	5.71	5.19	4.97
	MRR (%)	2.15	1.93	1.96
	ILD	1.24	1.28	1.32
Tmall	Recall (%)	14.57	14.43	14.32
	MRR (%)	7.50	7.45	7.43
	ILD	0.84	0.91	0.95



(a) $\lambda_{Dis} = 0.0$. **(b)** $\lambda_{Dis} = 0.5$ **(c)** $\lambda_{Dis} = 1.0$

Figure 4: Weight distributions of multiple intents with different values of λ_{Dis}.

From Figure 4, it is obvious when $\lambda_{Dis} = 0.0$, the three intents share similar attention weights distributions, which fails to extract this user's different intents and thus leads to worse performance in terms of diversity than that when $\lambda_{Dis} = 0.5$. As λ_{Dis} changes from 0.0 to 1.0, the differences between the three intents become more distinct. To sum up, the IIM module can effectively capture different latent intents with a disagreement regularization loss, as indicated by various weights for items in a sequence.

6.4 Case study

In this subsection, we show an example from the test set of ML100K to illustrate the different recommendation results by IDSR and NARM; see Figure 5.

Figure 5 (top) shows 7 movies that the user watched recently and the top 5 recommendations generated by IDSR and NARM, respectively. The ground truth item is marked with a red box. According to the user's historical views, we see that the user likes Children and Comedy recently. But the user also shows interest in Adventure, Animation, Action, Crime, Drama, Romance and Thriller. The items recommended by NARM are mainly in the Children genre, e.g., cartoon movies, which is close to the recent intents of this user. In contrast, IDSR accommodates multiple intents and diversifies the list of recommended movies with Drama, Crime, Romance and Thriller. IDSR also recognizes the most important intent and gives a high rank to the ground truth movie. This confirms that IDSR cannot only mine users' multiple intents, but generate a diversified list of recommended items to cover those intents.

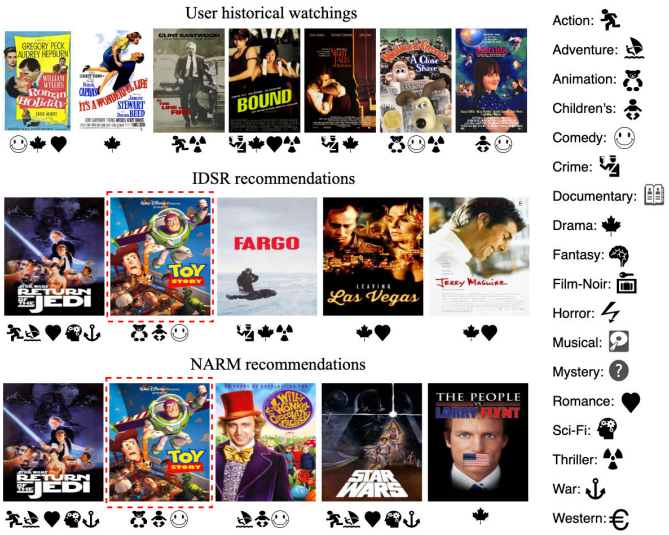


Figure 5: An example of recommendation results generated by IDSR and NARM.

7 CONCLUSION AND FUTURE WORK

In this paper, we have proposed the *intent-aware diversified sequential recommendation* (IDSR) model to improve diversification for sequential recommendation (SR). We have devised an implicit intent mining (IIM) module to capture users' multiple intents and an intent-aware diversity promoting (IDP) decoder to generate a diverse list of recommendations covering those intents. We have also designed an intent-aware diversity promoting (IDP) loss to supervise the model to simultaneously consider accuracy and diversification during training. We have conducted experiments on four datasets and have found that IDSR significantly outperforms the state-of-the-art baselines in terms of recommendation diversity while maintaining competitive accuracy scores. In addition, we have discussed the impact of the trade-off parameter and the number of intents as well as the disagreement regularization on our model's performancee, and included a case study to compare the items recommended by IDSR vs. those recommended by the baseline model.

As to future work, we plan to apply IDSR to other recommendation scenarios, e.g., shared-account recommendations, where the observed behavior may be generated by multiple users with more distinct intents [17, 28]. We also hope to improve the recommendation accuracy by incorporating other useful SR models into IDSR [39, 44]. In IDSR, there is a trade-off parameter controlling the balance between accuracy and diversity, i.e., λ, which needs to be pre-defined. This is a one-fits-all method that provides recommendations to all users with a constant accuracy-diversity balance. However, individuals have different needs for diversity, thus it is important to provide recommendations with an adaptive degree of diversity [10, 48]. We aim to investigate how to learn the trade-off parameter from users' behavior so as to address this need.

CODE AND DATA

To facilitate reproducibility of the results in this paper, we are sharing the code and data used to obtain those results at https://bitbucket.org/WanyuChen/idsr/.

ACKNOWLEDGMENTS

This research was partially supported by the China Scholarship Council and the National Natural Science Foundation of China under No. 61702526. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

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 Computing and Applications* 27, 5 (2016), 1207–1229.

[2] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Trans. Knowledge and Data Engineering.* 17, 6 (2005), 734–749.

[3] Azin Ashkan, Branislav Kveton, Shlomo Berkovsky, and Zheng Wen. 2015. Optimal Greedy Diversity for Recommendation. In *IJCAI '15.* 1742–1748.

[4] Betru Basiliyos, Tilahun, Onana Charles, Awono, and Batchakui Bernabe. 2017. Deep Learning Methods on Recommender System: A Survey of State-of-the-art. *International Journal of Computer Applications* 162, 10 (2017), 17–22.

[5] Jaime Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. In *SIGIR '98.* 335–336.

[6] Sotirios P. Chatzis, Panayiotis Christodoulou, and Andreas S. Andreou. 2017. Recurrent Latent Variable Networks for Session-Based Recommendation. In *DLRS '17.* 38–45.

[7] Laming Chen, Guoxin Zhang, and Hanning Zhou. 2018. Fast Greedy MAP Inference for Determinantal Point Process to Improve Recommendation Diversity. In *NIPS '18.* 5627–5638.

[8] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential Recommendation with User Memory Networks. In *WSDM '18.* 108–116.

[9] Peizhe Cheng, Shuaiqiang Wang, Jun Ma, Jiankai Sun, and Hui Xiong. 2017. Learning to Recommend Accurate and Diverse Items. In *WWW '17.* 183–192.

[10] Tommaso Di Noia, Jessica Rosati, Paolo Tomeo, and Eugenio Di Sciascio. 2017. Adaptive Multi-attribute Diversity for Recommender Systems. *Information Sciences* 382-383 (2017), 234 – 253.

[11] Xavier Glorot and Yoshua Bengio. 2010. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *AI&Statistics '10.* 249–256.

[12] Ruining He and Julian McAuley. 2016. Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation. In *ICDM '16.* 191–200.

[13] Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yugang Jiang, and Tatseng Chua. 2018. NAIS: Neural Attentive Item Similarity Model for Recommendation. *IEEE Trans. Knowledge and Data Engineering* 30, 12 (2018), 2354–2366.

[14] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. In *CIKM '18.* 843–852.

[15] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *ICLR '16.* 1–10.

[16] Dietmar Jannach. 2018. Keynote: Session-Based Recommendation – Challenges and Recent Advances. In *KI 2018: Advances in Artificial Intelligence.* 3–7.

[17] Jyun Yu Jiang, Cheng Te Li, Yian Chen, and Wei Wang. 2018. Identifying Users Behind Shared Accounts in Online Streaming Services. In *SIGIR '18.* 65–74.

[18] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *ICDM '18.* 197–206.

[19] Yejin Kim, Kwangseob Kim, Chanyoung Park, and Hwanjo Yu. 2019. Sequential and Diverse Recommendation with Long Tail. In *IJCAI '19.* 2740–2746.

[20] Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (2014).

[21] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[22] Alex Kulesza and Ben Taskar. 2012. *Determinantal Point Processes for Machine Learning.* Now Publishers Inc.

[23] Matev Kunaver and Toma Porl. 2017. Diversity in Recommender Systems A Survey. *Know.-Based Syst.* 123, C (2017), 154–162.

[24] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *CIKM '17.* 1419–1428.

[25] Jian Li, Zhaopeng Tu, Baosong Yang, Michael R. Lyu, and Tong Zhang. 2018. Multi-Head Attention with Disagreement Regularization. In *EMNLP '18.* 2897—2903.

[26] Shangsong Liang, Emine Yilmaz, Hong Shen, Maarten de Rijke, and W Bruce Croft. 2017. Search Result Diversification in Short Text Streams. *ACM Trans. Information Systems* 36, 1 (2017), 8.

[27] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. In *KDD '18.* 1831–1839.

[28] 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 *SIGIR '19.* 685–694.

[29] Richard Mccreadie, Rodrygo L T Santos, Craig Macdonald, and Iadh Ounis. 2018. Explicit Diversification of Event Aspects for Temporal Summarization. *ACM Trans. Information Systems* 36, 3 (2018), 25.

[30] Lijing Qin and Xiaoyan Zhu. 2013. Promoting Diversity in Recommendation by Entropy Regularizer. In *IJCAI '13.* 2698–2704.

[31] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. *Comput. Surveys* 51, 4, Article 66 (2018), 36 pages.

[32] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks. In *Recsys '17.* 130–137.

[33] Lakshmanan Rakkappan and Vaibhav Rajan. 2019. Context-Aware Sequential Recommendations with Stacked Recurrent Neural Networks. In *WWW '19.* 3172–3178.

[34] 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 *AAAI '19.* 4806–4813.

[35] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing Personalized Markov Chains for Next-basket Recommendation. In *WWW '10.* 811–820.

[36] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *WWW '01.* 285–295.

[37] Chaofeng Sha, Xiaowei Wu, and Junyu Niu. 2016. A Framework for Recommending Relevant and Diverse Items. In *IJCAI '16.* 3868–3874.

[38] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *CIKM '19.* 1441–1450.

[39] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *WSDM '18.* 565–573.

[40] 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 *NIPS '17.* 5998–6008.

[41] 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 *SIGIR '19.* 345–354.

[42] 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 *IJCAI '19.* 3771–3777.

[43] Qiong Wu, Yong Liu, Chunyan Miao, Yin Zhao, Lu Guan, and Haihong Tang. 2019. Recent Advances in Diversified Recommendation. *arXiv preprint arXiv:1905.06589* (2019).

[44] 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 *WWW '19.* 3398–3404.

[45] Y. Yao, F. Shen, J. Zhang, L. Liu, Z. Tang, and L. Shao. 2019. Extracting Privileged Information for Enhancing Classifier Learning. *IEEE Trans. Image Processing* 28, 1 (Jan 2019), 436–450.

[46] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential Recommender System based on Hierarchical Attention Networks. In *IJCAI '18.* 3926–3932.

[47] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A Dynamic Recurrent Model for Next Basket Recommendation. In *SIGIR '16.* 729–732.

[48] Ting Yu, Junpeng Guo, Wenhua Li, Harry Jiannan Wang, and Ling Fan. 2019. Recommendation with Diversity: An Adaptive Trust-aware Model. *Decision Support Systems* 123 (2019), 113073.

[49] Mi Zhang and Neil Hurley. 2008. Avoiding Monotony: Improving the Diversity of Recommendation Lists. In *Recsys '08.* 123–130.

[50] 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 Recognition* 56 (2016), 1–15.

[51] Andrew Zimdars, David Maxwell Chickering, and Christopher Meek. 2001. Using Temporal Data for Making Recommendations. In *UAI '01.* 580–588.