UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

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

Chen, W.; Ren, P.; Cai, F.; de Rijke, M.

DOI
10.48550/arXiv.1908.10171

Publication date
2019

Document Version
Final published version

Link to publication

Citation for published version (APA):
Chen, W., Ren, P., Cai, F., & de Rijke, M. (2019). *Improving End-to-End Sequential Recommendations with Intent-aware Diversification*. (v1 ed.) ArXiv. https://doi.org/10.48550/arXiv.1908.10171

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

*UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)*

Download date:12 Oct 2026

arXiv:1908.10171v1 [cs.IR] 27 Aug 2019

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

**Wanyu Chen**[1, 2] **Pengjie Ren**[2] **Fei Cai**[1] **Maarten de Rijke**[2]
[1] National University of Defense Technology, Changsha, China
[2] University of Amsterdam, Amsterdam, The Netherlands
wanyuchen, caifei@nudt.edu.cn, p.ren, M.deRijke@uva.nl

**Abstract**

Sequential Recommendations (SRs) that capture users' dynamic intents by modeling user sequential behaviors can recommend closely accurate products to users. Previous work on SRs is mostly focused on optimizing the recommendation accuracy, often ignoring the recommendation diversity, even though it is an important criterion for evaluating the recommendation performance. Most existing methods for improving the diversity of recommendations are not ideally applicable for SRs because they assume that user intents are static and rely on post-processing the list of recommendations to promote diversity. We consider both recommendation accuracy and diversity for SRs by proposing an end-to-end neural model, called Intent-aware Diversified Sequential Recommendation (IDSR). Specifically, we introduce an Implicit Intent Mining (IIM) module into SRs to capture different user intents reflected in user behavior sequences. Then, we design an Intent-aware Diversity Promoting (IDP) loss to supervise the learning of the IIM module and force the model to take recommendation diversity into consideration during training. Extensive experiments on two benchmark datasets show that IDSR significantly outperforms state-of-the-art methods in terms of recommendation diversity while yielding comparable or superior recommendation accuracy.

## 1 Introduction

Conventional recommendation methods often assume that user intents are static; they ignore the dynamic and evolving characteristics of user behavior. Sequential Recommendations (SRs) have been introduced to address this issue; they aim to predict the next item(s) by modeling the sequence of a user's previous behavior (Quadrana, Cremonesi, and Jannach 2018).

Early studies into SRs are mostly based on Markov Chains (MC) (Rendle, Freudenthaler, and Schmidt-Thieme 2010). Due to the unmanageable state space issue of MCs (Hidasi et al. 2016a), Recurrent Neural Network (RNN) or Transformer based neural models have attracted a lot of attentions (Kang and McAuley 2018). A number of studies have investigated various factors that might influence SR performances, e.g., personalization (Quadrana et al. 2017), repeat consumption (Ren et al. 2019), context (Rakkappan and Rajan 2019), etc. However, these methods focus on improving recommendation accuracy only, which might have the risk of over-specialization, i.e., the recommended items are super homogeneous.

**Figure 1: A example showing sequential recommendation with (bottom) and without (top) diversification.**

This is problematic considering that users usually have multiple intents. For example, as shown in Fig. 1, although the user shows most interest in cartoon movies from her/his historic watching behaviors, s/he also watches family and action movies occasionally. A better recommendation strategy should provide a diverse recommendation list to satisfy all these intents. In the case of Fig. 1, we should recommend a list containing action, cartoon and family movies simultaneously instead of only cartoons. Besides, sometimes, user intents are exploratory which means they do not have a specific intent in mind. Thus homogeneous recommendation lists cannot satisfy such users and they easily get bored with the low diverse recommendation lists (Sha, Wu, and Niu 2016).

Diversification has been well studied in conventional recommendations (Wu et al. 2019) as well as Web search (Abid et al. 2016). Current approaches for diversified recommendation mainly focus on how to re-rank the items based on a certain diversity metric with general recommendation models. However, they are not ideally applicable for SRs for two reasons. First, some of them assume user intents are static and require that user intents are prepared beforehand, which is unrealistic in most SR scenarios (Ashkan et al. 2015; Chen, Zhang, and Zhou 2018). Second, most of them 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 recommendation list based on both the item scores and

some implicit/explicit diversity metrics (Wu et al. 2019; Kunaver and Porl 2017). Because the recommendation models are not aware of diversity during learning and it is hard to design ideal diversity strategies for different recommendation models, these methods are generally inferior and far from satisfactory.

To address the above issues, we take into account both recommendation accuracy and diversity, and propose an end-to-end neural model, namely Intent-aware Diversified Sequential Recommendation (IDSR), for SRs. Generally, IDSR employs an Implicit Intent Mining (IIM) module to automatically capture multiple latent user intents reflected in user behavior sequences, and directly generate accurate and diverse recommendation lists w.r.t the latent user intents. In order to supervise the learning of the IIM module and force the model to take recommendation diversity into consideration during training, we design an Intent-aware Diversity Promoting (IDP) loss which evaluates recommendation accuracy and diversity based on the whole generated recommendation lists. Specifically, a sequence encoder is first used to encode the user behaviors into representations. Then, the IIM module employs multiple attentions to mine user intents with each attention capturing a particular latent user intent. Finally, an intent-aware recommendation decoder is used to generate a recommendation list by selecting one item at a time. Especially, when selecting the next item, IDSR also takes the already selected items as input so that it can track the satisfaction degree of each latent user intent. During training, we devise the IDP loss to instruct IDSR to learn to mine and track user intents. All parameters are learned in an end-to-end back-propagation training paradigm within a unified framework. We conduct extensive experiments on two benchmark datasets. The results show that IDSR outperforms the state-of-the-art baselines on two publicly available datasets in terms of both accuracy metrics, i.e., Recall and MRR, and diversity metric, ILD.

Our contributions can be summarized as follows:

- We propose an Intent-aware Diversified Sequential Recommendation (IDSR) method, which is the first end-to-end neural framework that considers diversification for SRs, to the best of our knowledge.
- We devise an Implicit Intent Mining (IIM) module to automatically mine latent user intents from user behaviors and an intent-aware recommendation decoder to generate diverse recommendation lists.
- We present an IDP loss to better supervise IDSR in terms of recommendation accuracy and diversity.
- We carry out extensive experiments and analyses on two benchmark datasets to verify the effectiveness of IDSR.

# 2 Related Work

## 2.1 Sequential recommendation

Traditional methods are mainly based on MCs (Zimdars, Chickering, and Meek 2001), which investigate how to extract sequential patterns to learn users' next preferences with probabilistic decision-tree models. Following this idea, He and McAuley (2016) fuse similarity models with MCs for SRs to solve sparse recommendation problems. However, MC-based methods only model local sequential patterns between adjacent interactions, which could not take the whole sequence into consideration.

Recently, RNNs have been devised to model variable-length sequential data. Hidasi et al. (2016b) 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. (2017) extend this idea and develop a hierarchical RNN structure which takes users' profile into account by considering cross-session information. Attention mechanisms have been applied to recommendation tasks to help models exploit users' preferences better (He et al. 2018). Li et al. (2017) propose a neural attentive session-based recommendation machine that takes the last hidden state from the session-based RNN as the sequential behavior, and uses the other hidden states of previous clicks for computing attention to capture users' current preferences in a given session. Xu et al. (2019) propose a novel Recurrent Convolutional Neural Network model to capture both long-term as well as short-term dependencies for sequential recommendation. Recently, a purely attention-based sequence-to-sequence model, Transformer, has achieved competitive performance on machine translation tasks (Vaswani et al. 2017b). Kang and McAuley (2018) introduce Transformer into SRs by presenting a two-layer Transformer model to capture users sequential behaviors. Sun et al. (2019) introduce a Bidirectional Encoder Representations from Transformers for sequential recommendation.

Although there are a number of studies for SRs, they only focus on accuracy of the recommendation list. None of the aforementioned studies has considered users' multiple intents and the diversification for SRs.

## 2.2 Diversified recommendation

Promoting the diversity for recommendation or search results has received increasing research attentions. The most representative implicit approach is Maximal Marginal Relevance (MMR) (Carbonell and Goldstein 1998). MMR represents relevance and diversity by independent metrics and uses the notion of marginal relevance to combine the two metrics with a tradeoff parameter. Sha, Wu, and Niu (2016) introduce a submodular objective function to combine relevance, coverage of users interests, and the diversity between items. Learning To Rank (LTR) has also been exploited to address diversification (Cheng et al. 2017). Cheng et al. (2017) 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 problem is that they all need diversified ranking lists as ground-truth for learning (Wu et al. 2019), which is usually unavailable in recommendations. Recently, Chen, Zhang, and Zhou (2018) propose to improve recommendation diversification through Determinantal Point Process (DPP) (Kulesza and Taskar 2012) with a greedy maximum a posterior inference algorithm.

All above methods achieve recommendation accuracy and diversity in two separate processes, i.e., training an offline recommendation model to score items in terms of accuracy

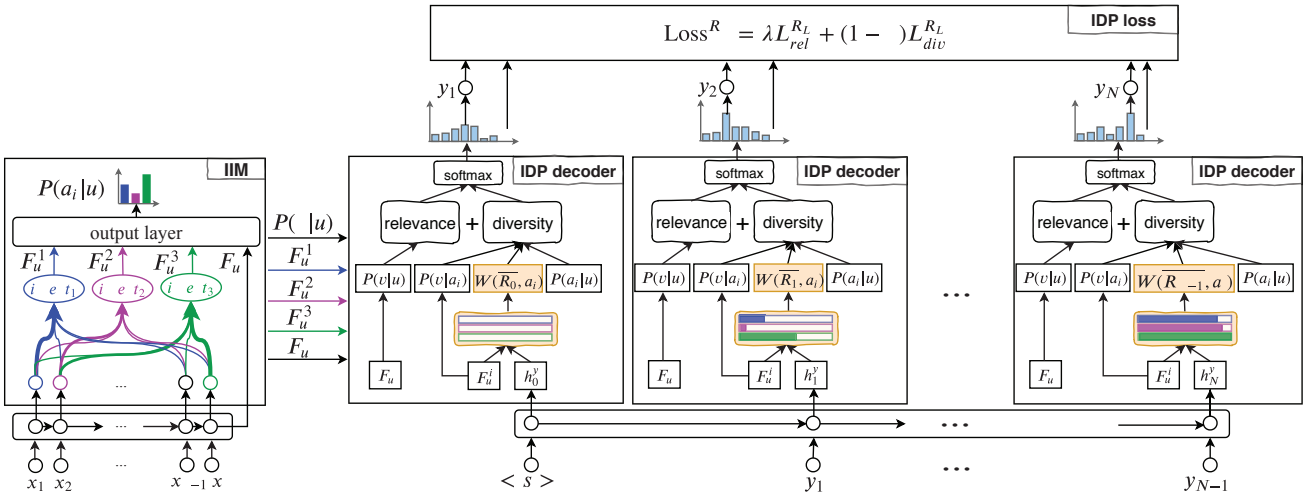



**Figure 2: Overview of IDSR. Blue, purple and green colors denote different user intents.**

and then re-ranking items by taking diversity into account. In addition, they are not suitable for SRs where users' sequential behaviors need to be considered.

## 3 Intent-aware Diversified Sequential Recommendation

### 3.1 Overview

Given a user $u$ and her/his behavior sequence $S_u$=$\{x_1, x_2, \ldots, x_T\}$ ($x_i$ is the item that $u$ interacts with, e.g., watched movies), our goal is to provide $u$ with a recommendation list $R_L$ for predicting her/his next interaction, of which the items are expected to be both relevant and diverse.

Different from 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 recommendation list $R_L$ by maximizing the satisfactory degree of all intents.

$$P(R_L \mid u, S_u) = \sum_{i=1}^{M} P(a_i \mid u) P(R_L \mid a_i, u, S_u), \quad (1)$$

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

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 $S(v)$.

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

where $v_t$ is the item to be selected at step $t$; $V$ is the set of all items; $R_{t-1}$ is the generated recommendation list 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 $t$. $S(v)$ returns the score of item $v$ by

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

Generally, it is a combination of the relevance score and diversification score balanced by a hyper-parameter $\lambda$. $P(v \mid u, S_u)$ is the relevance score reflecting the interest of $u$ on v. $P(a_i \mid u)$ is the probability of $u$ with the intent $a_i$. $P(v \mid a_i)$ is the satisfactory degree of $v$ to $a_i$. $W(\overline{R_{t-1}}, a_i)$ denotes the likelihood that the already generated recommendation list $R_{t-1}$ cannot satisfy $a_i$.

We propose an end-to-end IDSR model to directly generate a diversified recommendation list upon Eq. 3. As shown in Fig. 2, IDSR consists of three modules: a Sequence encoder, an Implicit Intent Mining (IIM) module and an Intent-aware Diversity Promoting (IDP) decoder. The sequence encoder encodes users' sequential behaviors into latent representations. Then, the IIM module is used to capture users' multiple latent intents reflected in the sequential behaviors. Finally, the IDP decoder is employed to generate a recommendation list w.r.t. Eq. 3. We devise an IDP loss to train IDSR which evaluates the whole recommendation list in terms of both recommendation accuracy and diversity. Note that there is no re-ranking involved in IDSR and both recommendation accuracy and diversity are jointly learned in an end-to-end way. Next, we introduce the separate modules.

### 3.2 Sequence encoder

Two commonly used technologies for sequence modeling are GRU and Transformer. We use both in our experiments (see §5) and find that IDSR shows better performance with

GRU. Thus here we use a GRU to encode $S_u$.

$$\begin{aligned} z_t &= \sigma(W_z[\mathbf{x_t}, h_{t-1}]) \\ r_t &= \sigma(W_r[\mathbf{x_t}, h_{t-1}]) \\ \hat{h_t} &= \tanh(W_h[\mathbf{x_t}, r_t \odot h_{t-1}]) \\ h_t &= (1 - z_t) \otimes h_{t-1} + z_t \odot \hat{h_t}, \end{aligned} \tag{4}$$

where $\mathbf{x_t}$ denotes the embedding of item $x_t$; $W_z$, $W_r$ and $W_h$ are weight parameters; $\sigma$ denotes the sigmoid function. The inputs of the encoder is the behavior sequence $S_u$=$\{x_1, x_2, \ldots, x_T\}$ and the outputs are hidden representations $\{h_1, h_2, \ldots, h_T\}$, where $h_i \in \mathbb{R}^{d_e}$. We stack those representations into matrix $H_S \in \mathbb{R}^{T \times d_e}$. Generally, we consider the last representation $h_T$ as a summary of the whole sequence (Li et al. 2017). Thus we set the global preference $F_u = h_T$.

### 3.3 IIM module

The IIM module is to evaluate the $P(a_i \mid u)$ in Eq. 3. Intuitively, a user's multiple intents can be reflected by different interactions in the sequential behaviors. Some interactions are more representative for a particular intent than the other interactions, e.g., the last two behaviors in Fig. 1 reflects the intent of watching cartoon movies. Motivated by this, we fuse a multi-intent attention mechanism with each attention to capture one particular intent. Specifically, IIM first projects $H_S$ and $F_u$ into $M$ spaces w.r.t. the latent intents respectively. Then, $M$ attention functions are employed in parallel to produce the user's intent-specific representations $\{F_u^1, F_u^2, \ldots, F_u^M\}$.

$$F_u^i = \text{Attention}(F_u W_i^Q, H_S W_i^K, H_S W_i^V) \tag{5}$$

where the projection matrices for intent $i$, i.e., $W_i^Q \in \mathbb{R}^{d_e \times d}$, $W_i^K \in \mathbb{R}^{d_e \times d}$ and $W_i^V \in \mathbb{R}^{d_e \times d}$, are learnable parameters. We use the scaled dot-product attention in this work (Vaswani et al. 2017a). After that, the importance of each intent, i.e., $P(a_i \mid u)$ in Eq. 3, can be calculated by

$$P(a_i \mid u) = \frac{\exp({F_u}^{\mathrm{T}} F_u^i)}{\sum_{j=1}^{M} \exp({F_u}^{\mathrm{T}} F_u^j)}. \tag{6}$$

### 3.4 IDP decoder

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

$$P(v \mid u, S_u) = \frac{\exp(\mathbf{v}^{\mathrm{T}} F_u)}{\sum_{j=1}^{|V|} \exp(\mathbf{v_j}^{\mathrm{T}} F_u)}. \tag{7}$$

Similarly, we model the relevance of $v$ to intent $a_i$ (i.e., $P(v \mid a_i)$ in Eq. 3) as follows:

$$P(v \mid a_i) = \frac{\exp(\mathbf{v}^{\mathrm{T}} F_u^i)}{\sum_{n=1}^{|V|} \exp(\mathbf{v_n}^{\mathrm{T}} F_u^i)}. \tag{8}$$

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 $\{h_1^y, h_2^y, \ldots, h_{t-1}^y\}$. Then we estimate the satisfactory of $R_{t-1}$ to each intent (i.e., $W(\overline{R_{t-1}}, a_i)$ in Eq. 3) by calculating the matching between $h_{t-1}^y$ and $F_u^i$ as follows:

$$\begin{aligned} W(\overline{R_{t-1}}, a_i) &= \frac{\exp(w_{t-1}^i)}{\sum_{j=1}^{M} \exp(w_t^j)}, \\ w_{t-1}^i &= -h_{t-1}^y W_y F_u^i. \end{aligned} \tag{9}$$

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

### 3.5 IDP loss

Since our goal is to generate a recommendation list which is both relevant and diverse, we design the IDP loss function to evaluate the whole generated list $R_L$:

$$\text{Loss}^{R_L} = \lambda L_{rel}^{R_L} + (1 - \lambda) L_{div}^{R_L}, \tag{10}$$

where $\lambda$ is a trade-off parameter, which is the same one as that in Eq 3, as they both control the contributions of accuracy and diversification. When $\lambda$ increases, we consider more about accuracy in both of IDP decoder and IDP loss. Given the output recommendation list from IDSR $R_L = \{y_1, y_2, \ldots, y_N\}$ and the ground-truth item $y^*$ (i.e., the next consumed item), $L_{rel}^{R_L}$ is defined as follows:

$$L_{rel}^{R_L} = \begin{cases} -\frac{1}{N} \sum_{t=1}^{N} t \cdot \mathbb{I}(y_t, y^*) \log \mathbf{p}_t^*, & y^* \in R_L \\ -\frac{1}{N} \sum_{t=1}^{N} \log \mathbf{p}_t^*, & y^* \notin R_L, \end{cases} \tag{11}$$

where $\mathbb{I}(y_i, y_t)$ is an indicator function that equals 1 if $y_i = y_t$ and 0 otherwise. $\mathbf{p}_t^*$ denotes the probability of the ground-truth item $y^*$ at $t$-th step when generating the recommendation list $R_L$. $L_{rel}^{R_L}$ encourages the positive recommendation list which contains the ground-truth item $y^*$ and punishes the negative ones otherwise. Note that $L_{rel}^{R_L}$ also takes the ranking position of the ground-truth item $y^*$ into consideration by weighting the loss with the position $t$.

To promote diversity, inspired by the result diversification in Web search (Agrawal et al. 2009), we define $L_{div}^{R_L}$ as the probability of each intent $a_i$ having at least one relevant item in $R_L$

$$L_{div}^{R_L} = -\sum_{i=1}^{M} P(a_i \mid u) \left(1 - \prod_{v \in R_L} (1 - P(v \mid a_i))\right),$$

where $P(a_i \mid u)$ and $P(v \mid a_j)$ are defined in Eq. 6 and Eq. 8 respectively.

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 seek to answer the following research questions:

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

**Table 1: Dataset statistics.**

| Dataset | ML100K | ML1M |
|---|---|---|
| Number of users | 943 | 6,040 |
| Number of items | 1,682 | 3,706 |
| Number of interactions | 100,000 | 1,000,209 |
| Number of item genres | 19 | 18 |
| Avg. number of genres per item | 1.7 | 1.6 |

(**RQ2**) Does IDSR outperform state-of-the-art baselines in terms of diversity?
(**RQ3**) What is the impact of different sequence encoders (i.e., GRU, Transformer) on IDSR?
(**RQ4**) How does the trade-off parameter $\lambda$ affect the performance of IDSR?
(**RQ5**) How does IDSR influence the recommendation results?

### 4.1 Datasets

To answer our research questions, we use two publicly available datasets for the experiments. Table 1 lists the statistics of the two 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 4,607,047 ratings for movies (Zhang, Yao, and Sun 2017).

We do not use the datasets as in (Quadrana et al. 2017) because there are only item ids. We cannot conduct diversity evaluation nor case study. We preprocess ML100K and ML1M for SR experiments with the following steps. First, we filter out users who have less than 5 interactions and the movies that are rated less than 5 times. Then, we sort the rated movies according to the "timestamp" field to get a sequence of behaviors for each user. Finally, we prepare each data sample by regarding the former 9 behaviors as input and the next behavior as output. For evaluation, we randomly divide the datasets into training (70%), validation (10%) and test (20%) sets. We make sure that all movies in the test set have been rated by at least one user in the training set and the test set contains the most recent behaviors which happened later than those in the training and validation sets.

### 4.2 Comparison methods

We select several traditional recommendation methods as well as recent state-of-the-art neural SR methods as baselines.

- **POP**: POP ranks items based on the number of interactions, which is a non-personalized approach (Adomavicius and Tuzhilin 2005).
- **FPMC**: FPMC is a hybrid model that combines MCs and collaborative filtering for SRs (Rendle, Freudenthaler, and Schmidt-Thieme 2010).
- **GRU4Rec**: GRU4Rec is an RNN-based model for SRs. GRU4Rec utilizes session-parallel mini-batches as well as a ranking-based loss function in the training process (Hidasi et al. 2016b).
- **HRNN**: HRNN is a hierarchical RNN for SRs based on GRU4Rec. It adopts a session-level RNN and a user-level RNN to model users' short-term and long-term preferences (Quadrana et al. 2017).

Because there is no previous work specific for diversified SRs, we construct three baselines **FPMC+MMR**, **GRU4Rec+MMR** and **HRNN+MMR** ourselves. Specifically, we first get the relevance scores $S(v)$ for each item with FPMC, GRU4Rec or HRNN. 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. We cannot choose the best $\theta$ relying solely on accuracy or diversity. To balance both, we set $\theta = 0.5$ in our experiments.

In addition, we consider two variants of our IDSR model:

- **IDSR**$_{trans}$: We use transformer to encoder users' behavior sequences in IDSR.
- **IDSR**$_{GRU}$: We use GRU to encoder users' behavior sequences in IDSR.

### 4.3 Evaluation metrics

For accuracy evaluation, we use Recall and MRR (Li et al. 2017; Liu et al. 2018); For diversity evaluation, we use Intra-List Distance (ILD) (Zhang and Hurley 2008):

- **Recall**: A primary metric which is used to evaluate the recall of the recommender system, i.e., whether the test item is contained in the recommendation list.
- **MRR**: A metric measures the ranking accuracy of the recommender system, i.e., whether the test item is ranked at the top of the list.
- **ILD**: A commonly used metric which measures the diversity of a recommendation list as the average distance between pairs of recommended items. ILD is defined as:

$$\text{ILD} = \frac{2}{|R_L|(|R_L|-1)} \sum_{(i,j) \in R_L} d_{ij}. \qquad (12)$$

We calculate the dissimilarity $d_{ij}$ between two movies based on Euclidean distance between the item genre vectors of movies (Ashkan et al. 2015).

### 4.4 Implementation details

We set the item embedding size and GRU hidden state sizes to 100. We use dropout with drop ratio $p = 0.1$. We initialize model parameters randomly using the Xavier method (Glorot and Bengio 2010). We optimize the model using Adam (Kingma and Ba 2014) 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 128. We test the model performance on the validation set for every epoch. The code used to run the experiments is available online[2].

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

[2] https://url.suppressed.for.anonymity

# 5 Results and Analysis

## 5.1 Performance in terms of accuracy

To answer RQ1, we examine the performance of IDSR and the baseline models in terms of Recall and MRR; see Table 2.

First, we can see that IDSR outperforms the traditional methods, i.e., POP and FPMC, as well as the neural-based methods, i.e., GRU4Rec and HRNN in terms of Recall and MRR. When the size of recommendation list changes from 10 to 20, the improvements of IDSR over the best baseline HRNN get increased. In detail, the improvements are 4.16% and 6.37% in terms of Recall@10 and Recall@20; 5.52% and 6.72% in terms of MRR@10 and MRR @20 on ML100K. Besides, the improvements of IDSR over HRNN in terms of MRR are larger than that of Recall. For instance, on the ML1M dataset, the improvements of IDSR over HRNN are 5.32% in terms of MRR@20 while 3.22% in terms of Recall@20. This shows that our model can not only boost the number of relevant items but the ranking of relevant items. It may be due to the fact that our IIM module can capture a user's multiple intents and their importances in his current preferences. Thus at the first steps in IDP decoder, IDSR can give high probabilities to relevant items.

Second, we note that after re-ranking with MMR, the performances of the baseline models drop a little bit in terms of both Recall and MRR. This indicates that although post-processing with MMR can improve the diversity of recommendation list, it might hurt the accuracy. Because most of the candidate items generated by the baseline models have similar genres, the diversity score for the relevant item may be much lower than irrelevant ones. This might lead to a situation that the irrelevant items have higher final scores than the relevant item, which results in a worse performance.

## 5.2 Performance in terms of diversity

To answer RQ2, we report the ILD scores on both datasets in Table 2. We can see that IDSR consistently outperforms all baselines. The improvements of IDSR over HRNN are 19.90% and 20.29% in terms of ILD@10 and ILD@20 on ML100K dataset, 57.76% and 50.85% on ML1M dataset. Compared with the post-processing baselines, our model outperforms the best baseline model, i.e., HRNN+MRR, by 7.39% and 7.97% in terms of ILD@10 and ILD@20 on ML100K dataset and 10.83% and 11.81% on ML1M dataset. Significant improvements of IDSR against the best performing baseline are observed with a paired $t$-test. This proves that our model is competitive in improving the diversity for sequential recommendation than post-processing methods. It may be because that the effectiveness of post-processing methods, e.g., MMR, can be impacted by the baseline models. When the candidate items generated by the sequential recommendation baselines are of similar genres, the performance of MMR method is limited to some extent.

## 5.3 Performance with different encoders

To answer RQ3, we test the performance of IDSR with different sequence encoders, i.e, GRU or Transformer. Table 3 shows the results.

IDSR shows a better performance with GRU than with Transformer in terms of all metrics on both datasets. We believe the reason is that Transformers rely on position embeddings to capture sequential information which are less effective than GRU, especially when Transformers are not pre-trained on a large-scale dataset. On the contrast, the recurrent nature of GRU is especially designed for sequence modeling, which means it needs fewer data to capture the sequential information. This can be proved by the fact that the improvements of $\text{DSR}_{GRU}$ over $\text{DSR}_{Trans}$ on ML1M are smaller than that on ML100K. For example, the improvements in terms of Recall@20 are 10.94% and 5.69% on ML100K and ML1M, respectively.

## 5.4 Influence of trade-off parameter $\lambda$

To answer RQ4, we investigate the impact of $\lambda$ on IDSR by ranging it from 0 to 1 with a step size of 0.1. The results are shown in Fig. 3.

We can see that the accuracy metrics, i.e., Recall@20 and MRR@20, show upward trends generally when $\lambda$ increases from 0 to 1. When $\lambda = 0$, IDSR shows the worst performance. However, a noticeable increase is observed when $\lambda$ changes from 0 to 0.1. It is because that $\lambda = 0$ means 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 $\lambda$ at around 0.5 on ML100K, while around 0.8 on ML1M dataset.

Regarding recommendation diversity, IDSR increases when $\lambda$ changes from 0 to 0.1 and then decreases from 0.4 (0.2) to 1 in terms of ILD@20 on both datasets. The best diversification performance of IDSR appears with a small value of $\lambda$, i.e., 0.2 on ML100K and 0.4 on ML1M. When $\lambda$ is set to 1, ILD@20 of IDSR drops significantly as it indicates that we do not consider diversification in our IDP loss. To conclude, Fig. 3 demonstrates that our designed IDP loss can boost the performance of IDSR when we take both of accuracy and diversification into consideration simultaneously.

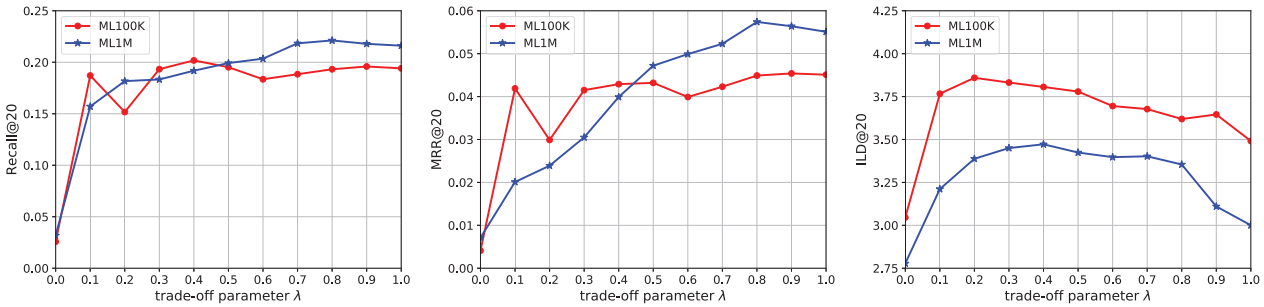
## 5.5 Case study

To answer RQ5, we select an example from the test set of ML100K to illustrate the different recommendation results by IDSR and HRNN in Fig. 4.

Fig. 4 shows 8 movies that the user watched recently and the top 5 recommendations generated by IDSR and HRNN models, respectively. The ground truth item is marked with red box. According to the user's historical watchings, we can tell that the user like Drama the most. But the user also shows interest in Comedy, Action, Thriller. HRNN recommends four Drama movies which only takes care of the main intent of this user. Differently, IDSR accommodates all intents and diversifies the recommendation list with Drama, Comedy, Action, Adventure and Thriller movies. Especially, IDSR also recognizes the most important intent and rank a Drama movie at the top. This confirms that IDSR can not only mine users' implicit intents, but also generate a diversified recommendation list to cover those intents.

**Table 2: Performance of recommendation models. The results produced by the best baseline and the best performer in each column are underlined and boldfaced, respectively. Statistical significance of pairwise differences of IDSR$_{GRU}$ vs. the best baseline is determined by a paired $t$-test ($^{\blacktriangle}$ for p-value $\leq$ .01, or $^{\triangle}$ for p-value $\leq$ .05).**

| | ML100K | | | | | | ML1M | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Recall | | MRR | | ILD | | Recall | | MRR | | ILD | |
| Model | @10 | @20 | @10 | @20 | @10 | @20 | @10 | @20 | @10 | @20 | @10 | @20 |
| POP | .0397 | .0804 | .0131 | .0166 | 3.330 | 3.401 | .0578 | .1121 | .0181 | .0277 | 2.518 | 2.681 |
| FPMC | .0411 | .1011 | .0155 | .0207 | 3.011 | 3.071 | .0731 | .1344 | .0201 | .0447 | 2.001 | 2.017 |
| GRU4Rec | .1082 | .1787 | .0321 | .0371 | 3.121 | 3.144 | .1101 | .1956 | .0399 | .0501 | 2.006 | 2.212 |
| HRNN | .1106 | .1897 | .0362 | .0402 | 3.148 | 3.164 | .1171 | .2143 | .0418 | .0545 | 2.045 | 2.224 |
| FPMC+MMR | .0397 | .0971 | .0121 | .0197 | 3.374 | 3.415 | .0701 | .1332 | .0199 | .0431 | 2.865 | 2.932 |
| GRU4Rec+MMR | .1037 | .1711 | .0311 | .0361 | 3.493 | 3.506 | .1043 | .1901 | .0383 | .0489 | 2.879 | 2.985 |
| HRNN+MMR | .1094 | .1857 | .0351 | .0396 | 3.515 | 3.525 | .1141 | .2101 | .0411 | .0532 | 2.911 | 3.000 |
| IDSR$_{GRU}$ | **.1152**$^{\blacktriangle}$ | **.2018**$^{\blacktriangle}$ | **.0382**$^{\blacktriangle}$ | **.0429**$^{\blacktriangle}$ | **3.775**$^{\blacktriangle}$ | **3.806**$^{\blacktriangle}$ | **.1203**$^{\triangle}$ | **.2212**$^{\triangle}$ | **.0431**$^{\triangle}$ | **.0574**$^{\triangle}$ | **3.227**$^{\blacktriangle}$ | **3.354**$^{\blacktriangle}$ |




**(a) Performance in terms of Recall@20.** **(b) Performance in terms of MRR@20.** **(c) Performance in terms of ILD@20.**

**Figure 3: Performance of IDSR on two datasets with the parameter $\lambda$ in Eq. (11) changing from 0 to 1.**

**Table 3: Performance of IDSR model with different sequence encoder (GRU or Transformer). The results produced by the best performer in each row are boldfaced.**

| Dataset | Metric | IDSR$_{GRU}$ | IDSR$_{Trans}$ |
|---|---|---|---|
| ML100K | Recall@20 | **.2018** | .1819 |
| | MRR@20 | **.0429** | .0385 |
| | ILD@20 | **3.806** | 3.635 |
| ML1M | Recall@20 | **.2212** | .2093 |
| | MRR@20 | **.0574** | .0529 |
| | ILD@20 | **3.354** | 3.313 |

## 6 Conclusion and Future Work

In this paper, we propose IDSR model to improve the diversification for SRs. We devise the IIM module to capture users' multiple intents and IDP decoder to generate a diversified recommendation list covering those intents. We also design an IDP loss to supervise the model to consider accuracy and diversification simultaneously during training. Our experimental results and in-depth analyses confirm the effectiveness of IDSR on two datasets.

As to future work, on the one hand, we plan to apply our model to other recommendation scenes, e.g., shared-account recommendations where the behaviors can be from multiple users with totally different intents. On the other hand, we hope to improve the recommendation accuracy by incorporating useful mechanisms from recent SR models into IDSR.

**Figure 4: An example of recommendation results generated by IDSR and HRNN.**

## References


[Abid et al. 2016] Abid, A.; Hussain, N.; Abid, K.; Ahmad, F.; Farooq, M. S.; Farooq, U.; Khan, S. A.; Khan, Y. D.; Naeem, M. A.; and Sabir, N. 2016. A survey on search results diversification techniques. *Neural Computing and Applications* 27(5):1207–1229.

[Adomavicius and Tuzhilin 2005] Adomavicius, G., and Tuzhilin, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering.* 17(6):734–749.

[Agrawal et al. 2009] Agrawal, R.; Gollapudi, S.; Halverson, A.; and Ieong, S. 2009. Diversifying search results. In *WSDM '09*, 5–14. New York, NY, USA: ACM.

[Ashkan et al. 2015] Ashkan, A.; Kveton, B.; Berkovsky, S.; and Wen, Z. 2015. Optimal greedy diversity for recommendation. In *IJCAI'15*, 1742–1748. AAAI Press.

[Carbonell and Goldstein 1998] Carbonell, J., and Goldstein, J. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *SIGIR '98*, 335–336. New York, NY, USA: ACM.

[Chen, Zhang, and Zhou 2018] Chen, L.; Zhang, G.; and Zhou, H. 2018. Fast greedy map inference for determinantal point process to improve recommendation diversity. In *NIPS'18*, 5627–5638. USA: Curran Associates Inc.

[Cheng et al. 2017] Cheng, P.; Wang, S.; Ma, J.; Sun, J.; and Xiong, H. 2017. Learning to recommend accurate and diverse items. In *WWW '17*, 183–192.

[Glorot and Bengio 2010] Glorot, X., and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AI&Statistics '10*, 249–256. Chia Laguna Resort, Sardinia, Italy: PMLR.

[He and McAuley 2016] He, R., and McAuley, J. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *International Conference on Data Mining*, 191–200. IEEE.

[He et al. 2018] He, X.; He, Z.; Song, J.; Liu, Z.; Jiang, Y.; and Chua, T. 2018. Nais: Neural attentive item similarity model for recommendation. *IEEE Transactions on Knowledge and Data Engineering* 30(12):2354–2366.

[Hidasi et al. 2016a] Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2016a. Session-based recommendations with recurrent neural networks. In *ICLR '16*.

[Hidasi et al. 2016b] Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2016b. Session-based recommendations with recurrent neural networks. In *ICLR'16*, 1–10.

[Kang and McAuley 2018] Kang, W., and McAuley, J. J. 2018. Self-attentive sequential recommendation. In *ICDM '18*, 197–206.

[Kingma and Ba 2014] Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

[Kulesza and Taskar 2012] Kulesza, A., and Taskar, B. 2012. *Determinantal Point Processes for Machine Learning*. Hanover, MA, USA: Now Publishers Inc.

[Kunaver and Porl 2017] Kunaver, M., and Porl, T. 2017. Diversity in recommender systems a survey. *Know.-Based Syst.* 123(C):154–162.

[Li et al. 2017] Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; and Ma, J. 2017. Neural attentive session-based recommendation. In *CIKM '17*, 1419–1428. New York, NY, USA: ACM.

[Liu et al. 2018] Liu, Q.; Zeng, Y.; Mokhosi, R.; and Zhang, H. 2018. Stamp: Short-term attention/memory priority model for session-based recommendation. In *KDD '18*, 1831–1839. New York, NY, USA: ACM.

[Quadrana et al. 2017] Quadrana, M.; Karatzoglou, A.; Hidasi, B.; and Cremonesi, P. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *RecSys '17*, 130–137. New York, NY, USA: ACM.

[Quadrana, Cremonesi, and Jannach 2018] Quadrana, M.; Cremonesi, P.; and Jannach, D. 2018. Sequence-aware recommender systems. *ACM Computing Surveys* 51(4):66:1–66:36.

[Rakkappan and Rajan 2019] Rakkappan, L., and Rajan, V. 2019. Context-aware sequential recommendations with stacked recurrent neural networks. In *The Web Conference*, 3172–3178.

[Ren et al. 2019] Ren, P.; Chen, Z.; Li, J.; Ren, Z.; Ma, J.; and de Rijke, M. 2019. Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In *AAAI '19*. AAAI.

[Rendle, Freudenthaler, and Schmidt-Thieme 2010] Rendle, S.; Freudenthaler, C.; and Schmidt-Thieme, L. 2010. Factorizing personalized markov chains for next-basket recommendation. In *WWW '10*, 811–820. New York, NY, USA: ACM.

[Sha, Wu, and Niu 2016] Sha, C.; Wu, X.; and Niu, J. 2016. A framework for recommending relevant and diverse items. In *IJCAI'16*, 3868–3874. AAAI Press.

[Sun et al. 2019] Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. *CoRR* abs/1904.06690.

[Vaswani et al. 2017a] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017a. Attention is all you need. *CoRR* abs/1706.03762.

[Vaswani et al. 2017b] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, Aidan Nand Kaiser, L. u.; and Polosukhin, I. 2017b. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, 5998–6008. Curran Associates, Inc.

[Wu et al. 2019] Wu, Q.; Liu, Y.; Miao, C.; Zhao, Y.; Guan, L.; and Tang, H. 2019. Recent advances in diversified recommendation. *CoRR* abs/1905.06589.

[Xu et al. 2019] Xu, C.; Zhao, P.; Liu, Y.; Xu, J.; S.Sheng, V. S.; Cui, Z.; Zhou, X.; and Xiong, H. 2019. Recurrent convolutional neural network for sequential recommendation. In *WWW '19*, 3398–3404. New York, NY, USA: ACM.

[Zhang and Hurley 2008] Zhang, M., and Hurley, N. 2008. Avoiding monotony: Improving the diversity of recommendation lists. In *RecSys '08*, 123–130. New York, NY, USA: ACM.

[Zhang, Yao, and Sun 2017] Zhang, S.; Yao, L.; and Sun, A. 2017. Deep learning based recommender system: A survey and new perspectives. *arXiv preprint arXiv:1707.07435*.

[Zimdars, Chickering, and Meek 2001] Zimdars, A.; Chickering, D. M.; and Meek, C. 2001. Using temporal data for making recommendations. In *UAI '01*, 580–588. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

View publication stats