Learning Diversity Attributes in Multi-Session Recommendations

1st Nassim Bouarour
CNRS, Univ. Grenoble Alpes
Grenoble, France

2st Idir Benouaret*
CNRS, Univ. Grenoble Alpes
Grenoble, France

3st Sihem Amer-Yahia
CNRS, Univ. Grenoble Alpes
Grenoble, France

***Abstract*—Diversity in recommendation has been studied extensively. It has been shown that maximizing diversity subject to constrained relevance yields high user engagement over time. Existing work largely relies on setting some attributes that are used to craft an item similarity function and diversify results. In this paper, we examine the question of learning diversity attributes. That is particularly important when users receive recommendations over multiple sessions. We devise two main approaches to look for the best diversity attribute in each session: the first is a generalization of traditional diversity algorithms and the second is based on reinforcement learning. We implement both approaches and run extensive experiments on a semi-synthetic dataset. Our results demonstrate that learning diversity attributes yields a higher overall diversity than traditional diversity algorithms. We also find that training policies using reinforcement learning is more efficient in terms of response time, in particular for high dimensional data.**

I. Introduction

Diversity in search and recommendation has been the topic of a multitude of research efforts [1]–[6]. Recent works have shown that diversity improves user satisfaction both in single sessions and multiple sessions [7]. To enforce the diversity of a set of items, one has to fix the set of attributes used to compute item similarity. In this paper, we argue that a single choice of attributes does not fit all users and all sessions, and it is more appropriate to learn diverse attributes in a personalized fashion. The question of *learning diversity* has received limited attention with only a focus on single sessions [8], [9]. In this work, we tackle the problem of learning diversity in single and multiple recommendation sessions.

We view recommendation as a multi-session process where a set of items is returned to a user in each session. Consider the case of Sydney, a user who listens to music during a trip. Sydney starts with a playlist of Bob Dylan's songs from different eras (60's, 70's, etc) and different genres (Folk, Rock, etc). After some time, she receives a less diversified playlist composed mainly of Rock music from the 70's interpreted by a small number of artists. That is followed by a similar set of Rock songs from the 80's. In the end, Sydney listens to a playlist containing mostly Folk songs from a variety of eras. The main observation is that attributes that yield the highest diversity differ across sessions since they are data- and user-dependent. Sydney would benefit from an automated system that is able to capture the diversity of attributes across different sessions and suggest to her a series of playlists that judiciously combines session diversities.

Diversity in recommendation matters as it increases user retention and decreases churn [10], but it can also evolve and vary. Traditional diversity algorithms fail to capture this change in diversity. For this reason, we propose to identify diversity attributes in each session. Given a set of N items that are relevant to a user, our goal is to find the k best items, in terms of relevance and diversity, to return in the first session followed by the k best for the second session, etc. Identifying diversity attributes could be done in two ways: either by running a traditional diversity algorithm such as MMR (Maximal Marginal Relevance) [11], [12] and SWAP [3], and finding the attributes that yield the best diversity in each session, or by relying on machine learning techniques to identify those attributes.

Our first contribution is to leverage MMR and SWAP to find the best diversity attribute in each session. Our adaptation consists in iterating over all available attributes and items in a session and returning the attributes that maximize the objective function of MMR or SWAP. The set of available items in each session is formed by the items that have not been returned in the previous sessions either because of their low relevance or because they did not contribute to diversity. The algorithm goes on until the number of items N or the number of desired sessions is completed. Our second contribution is to

* Now at LRE-Laboratoire de Recherche de l'EPITA

learn the attributes of diversity in each session. Given the lack of logs, we propose to leverage Reinforcement Learning (RL) to train an agent based on a reward that reflects the best diversity attained in each session. The output of the training is a policy that generalizes single-session diversity and maximizes the underlying objective function across multiple sessions.

In recent works, RL models showed promising results when applied to learning diversity [13], [9]. The application of RL models allows predicting long-term goals that are important in a multi-session recommendation problem. Our third contribution is to adapt SMORL [9], a state-of-the-art RL solution for recommendation diversity, to take into account multi-session diversity. The SMORL model is composed of two heads, the self-supervised and RL regularizer. The former head is a fully connected layer that plays the role of a traditional recommender. It ranks all items by predicting their relevance and is trained using a cross-entropy loss. The latter is an RL head which modifies the initial ranking of items as they are trained simultaneously. It learns the Q-function using the Scalarized Deep Q-learning (SDQL) [14]. We extend this model with another fully connected layer that allows us to identify the right attribute that maximizes diversity by learning the Q-function using DQN [15].

Experiments. We use a real-world dataset, a merge between MovieLens 10M and IMDb datasets, containing 5 real attributes to generate a semi-synthetic dataset by augmenting it with additional simulated attributes. In summary, we found that our baselines are the best performers in terms of diversity regardless of the number of attributes or the number of sessions while our RL-based approach is the best in terms of accuracy. Additionally, the RL variant is the best time performer, especially for a high number of attributes.

II. Related Work

Our work is related to research on sequential and session-based recommendations, leveraging Reinforcement Learning (RL) for recommendation and diversity in recommendations, and multi-attribute diversity in recommendations.

Sequential and session-based recommendations. In sequential recommendations [16]–[18], the order in which items are consumed impacts the prediction of the next items. As reported in [19], session-based approaches can be divided into three classes. The conventional class relies on traditional data mining and machine learning models like KNN [20], Association Rules [21] that build a recommended session where the conditional probability of items is greater than a predefined confidence threshold, and Markov Chains [22] where the matrix of transition probabilities is factorized to capture unseen transitions. The second class is based on Matrix Factorization [23]. These methods map the interactions into low-dimensional latent representations of users and items that are used to build subsequent recommendation sessions. These previous classes only capture easy dependencies between adjacent items. For this reason, methods based on Deep Learning emerged to model complex interaction sequences. For instance, Hidasi et al. [24] proposed an RNN-based model using gated recurrent units (GRU) to model the order and dependencies between items in each session. In [25], the authors proposed a solution where the recent items are embedded into a matrix considered as an image that is used to capture sequential patterns and general preferences using convolution filters (CNN).

RL for Recommendation. Many works on recommender systems using RL started to emerge. One of the off-policy recommenders is [26]. The authors developed a model that has as a reward the activeness of the user as well as its action (click or no). Zhao et al. [27] extended that previous work by using, in addition to positive feedback, negative ones. They also designed a session-based RL recommendation model that optimizes the display of items on a web page [28]. Other works [29], [30] use a policy method based on REINFORCE [31] to directly optimize the policy instead of estimating the Q-value.

Model-based RL methods, which are alternatives to the previous methods called Model-Free methods, were also used in the context of recommendations [32], [33]. These approaches build a model to simulate and approximate the environment. Xin et al. [34] introduced a self-supervised approach that has two levels of output layers. The first one, called head, performs a next item recommendation while the second, based on RL, acts as a regularizer to fine-tune the recommendations. Their results show that the model outperforms other RL-based recommender methods.

These previous works design models that optimize utility and accuracy while in our work we build a model that additionally leverages diversity.

RL for Recommendation Diversity. Hansen et al. [13] proposed a simple RL ranker that samples items to produce a ranked list of diverse items. Results showed that the RL ranker outperforms traditional diversity algorithms. Another diversity-promoting RL model was developed by Liu et al. [35]. The work relies on an actor-critic algorithm to generate item relevance, user preferences, and a fixed matrix that captures the similarity between items. Their results show that D^{2}RL yields better results than matrix factorization and contextual bandits. Finally, Stamenkovic et al. [9] developed SMORL that extends the work of [34] adding two more objectives, diversity and novelty, in addition to relevance to produce recommendations. More precisely, they used the same head layer but they expanded the RL regularizer to multiple objectives. Their results show that SMORL produces more diverse recommendations while maintaining or improving accuracy compared to state-of-the-art RL models.

All these works leverage RL to solve the single-session diversity problem and they don't learn the best attribute that maximizes diversity.

Optimizing diversity in multiple sessions is a recent research topic with a focus on optimizing intra- and inter-session diversities, leading to four bi-objective optimization problems [7].

The difference with our work is that diversity attributes are given and not learned.

Multi-attribute Diversity in Recommendations. The most relevant work on multi-attribute recommendation diversity relies on classifying users based on their interest in diversity for each attribute [36], [37]. As a result, each user may belong to different classes for different attributes. A user-dependent weight is then manually assigned to each attribute based on the resulting classes. This weight is used to craft a personalized similarity function for each user that computes the similarity between items as the weighted average of all similarities of all attributes. This function is then leveraged by MMR [11], a well-known diversity approach.

This work learns a diversity function based on multiple attributes as opposed to a single attribute. However, it is only applicable in a single session while our work relies on RL to learn a policy applicable to multiple sessions.

III. Data Model and Goal

Let $\mathcal{U}$ denote the set of users and $\mathcal{I}$ the set of items that any user $u \in \mathcal{U}$ can choose from. An item $i \in \mathcal{I}$ is represented with a vector of attributes $< att_1, att_2, .., att_p >$ drawn from a set of attributes $\mathcal{A}$. For instance, in the music domain, items can be represented as < artist, genre, release date>, and a song may be represented as < *Pink Floyd*, *Rock*, *1979* >. We assume that we are given a distance $d : \mathcal{I} \times \mathcal{I} \times \mathcal{A} \rightarrow \mathbb{R}^+$ that reflects dissimilarity, i.e., the diversity of two items in $\mathcal{I}$ with respect to an attribute att. We use $d_{att}(i, j)$ to reflect the diversity of the two items i, j on attribute att.

Lately, recommendation systems relied on embeddings to encode latent representations of items [10], [13]. Once embeddings are computed, items that are strongly related to each other will have close representations in the item embedding space. To produce such an embedding space we follow a methodology that is used for Spotify [10]. It consists of training a Word2Vec algorithm [38] on user-item interactions. Traditionally, Word2Vec is used in natural language processing to embed words from a corpus of documents. In our case, we used it to produce item embeddings for each attribute. To train the Word2Vec models, we map each item to a numerical index and use as context the items sharing similar attribute values. The vocabulary size is represented by the number of items. For numerical attributes, we compute similarity between items using Cosine similarity. For categorical attributes, two items are similar if they share the same value. In the case where the number of similar items exceeds the context size, we choose items with similar attributes randomly.

To compute the diversity of a set of items in a given session S with respect to an attribute $att \in \mathcal{A}$, we leverage the widely used intra-list-distance measure [2] that computes the diversity of items in a session as the average pairwise distance between items in the session. More formally

$$div_{att}(S) = \frac{\sum_{i \in S} \sum_{j \in S \setminus \{i\}} d_{att}(i, j)}{|S|(|S| - 1)} \tag{1}$$

We can now state our goal: Given user u and a set of l sessions, recommend the k most diverse unseen items in each session. We aim to maximize the diversity of each session by finding the attribute that yields the highest diversity. More formally, in each session S, we look for an attribute att, s.t.:

$$argmax_{att}\ div_{att}(S),\ |S| = k$$

IV. Algorithms

To solve our problem, we devise two main approaches to look for the best diversity attribute in each session: the first is a generalization of traditional diversity algorithms (MMR in Section IV-A and SWAP in Section IV-B) and the second is based on reinforcement learning to learn diversity attributes in several sessions (Section IV-C).

A variety of diversity algorithms have been developed for recommendation. In this work, we leverage SWAP [3] and MMR [11], two common diversity algorithms, to develop our baselines. We adapt SMORL, a state-of-the-art RL architecture [9] for multi-attribute recommendation.

A. MMR Adaptation

A widely used approach to combine relevance and diversity is the greedy algorithm MMR [11], [12]. Given a set S of the k most relevant items to recommend to a user, MMR selects at each step a new candidate item i^* that maximizes a linear combination of its relevance and the gain in diversity that is achieved according to already selected items. More formally, it chooses i^* s.t.

$$i^* = \arg\max_{i \in \mathcal{I}} \left((1-\alpha) \cdot rel(i) + \alpha \min_{j \in S} d(i,j) \right)$$

where $\alpha \in [0,1]$ is a parameter that tunes the relative importance of each relevance and diversity. Higher values of α mean more importance is put on diversifying the resulting recommendation set.

Algorithm 1 illustrates our adaptation of MMR which consists in iterating over all attributes in $\mathcal{A}$ to find the one yielding the highest diversity.

Algorithm 1: Multi-attribute MMR

Input: user u, set of items $\mathcal{I}$, set of attributes $\mathcal{A}$, α, # recommendations k, # sessions l
Output: l-session recommendations

$\mathcal{X} \leftarrow \emptyset$
for $j \leftarrow 1$ **to** l **do**
  $C \leftarrow [\ Items\ in\ relevance\ order\ to\ u\ ]$
  $divs\_A \leftarrow []$ (To keep the result set of each attribute)
  foreach att *in* $\mathcal{A}$ **do**
    $S_j \leftarrow C[0]$
    while $|S_j| < k$ **do**
      $i^* = \arg\max_{i \in C \setminus S_j} ((1-\alpha) \cdot rel(i) + \alpha \min_{j \in S} div_{att}(i,j))$
      $S_j \leftarrow S_j \cup \{i^*\}$
    end
    $divs\_A.append(S_j)$
  end
  $S^* \leftarrow$ Get the set with the highest diversity in $divs\_A$
  $C \leftarrow C \setminus S^*$
  $\mathcal{X}.append(S^*)$
end
return $\mathcal{X}$

B. SWAP Adaptation

We propose algorithm 2 based on SWAP, a re-ranking approach that is divided into two steps. First, a recommendation algorithm is used to predict the relevance of unseen items to a user u (Line 3). A top-k list of items S is selected (Line 4). Secondly, for each attribute in $\mathcal{A}$ (Line 6), a recommended list (S_{att}) is computed by a standard SWAP algorithm using the same initial S where items that contribute the least to diversity are replaced with ones that maximize it (Lines 8-16). The list S_{att} with the highest diversity is selected as the session to recommend (Line 19).

Algorithm 2: Multi-attribute SWAP

Input: a user u, a set of items $\mathcal{I}$, a set of attributes $\mathcal{A}$, # recommendations k, # sessions l
Output: l-session recommendation

1 $\mathcal{X} \leftarrow \emptyset$
2 **for** $j \leftarrow 1$ **to** l **do**
3   $C \leftarrow [\ Items\ in\ relevance\ order\ to\ u\ ]$
4   $S \leftarrow Topk(C)$
5   $divs\_A \leftarrow []$ (To keep the result set of each attribute)
6   **foreach** att *in* $\mathcal{A}$ **do**
7     $S_{att} \leftarrow S$; $m = 1$
8   **while** $m < k$ **do**
9       $pos = k + m$; $m = m + 1$
10   **for** i *in* S_{att} **do**
11   **if** $div_{att}(S_{att}) < div_{att}((S_{att} - \{i\}) \cup C[pos])$ **then**
12           $S_{att}.remove(i)$
13           $S_{att}.add(C[pos])$
14   **end**
15   **end**
16   **end**
17     $divs\_A.append(S_{att})$
18   **end**
19   $S^* \leftarrow$ Get the set with the highest diversity in $divs\_A$
20   $C \leftarrow C \setminus S^*$
21   $\mathcal{X}.append(S^*)$
22 **end**
23 **return** $\mathcal{X}$

C. Learning Multi-Attribute Diversity

We formalize our problem as a Markov Decision Process (MDP) in which the agent interacts with the environment represented by all users. We define the key components of the MDP represented by the tuple ($\mathcal{S}$, $\mathcal{A}$, $\mathcal{P}$, $\mathcal{R}$, γ) as follows:

- State Space ($\mathcal{S}$): It describes the user state at time t. A state s_t is represented by an embedding vector that summarizes the last session, defined by the last k items the user interacted with. More formally, $s_t = F_{emb}(c_{t-1})$ where c_{t-1} is the $(t-1)^{th}$ session. F_{emb} should capture the connections between the different items within the session as well as their order which makes it different from having as input the union of all items of the session.
- Action Space ($\mathcal{A}$): An action permits the transition between two consecutive states. In this work, we consider two types of actions: Choosing the items that define the next session recommended to the user and selecting the attribute for which the diversity of the next session is maximized. More formally, $a_t = (c_t, att_t)$ where c_t is the t^{th} session and att_t is the selected attribute.
- State transition probability ($\mathcal{P}$): $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$ is the probability $p(s_{t+1}|s_t, a_t)$ of transition from s_t to s_{t+1} when the agent selects the action a_t.
- Reward ($\mathcal{R}$): $\mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the instant reward of taking an action a_t at state s_t. More formally, $r(s_t, a_t) = div_{att_t}(c_t)$ where div_{att_t} is the intra-diversity of the selected session c_t using the selected attribute att_t.
- $\gamma \in [0, 1]$ which represents the discount factor for future rewards. If $\gamma = 0$ the agent ignores all future rewards and considers only the immediate one. If $\gamma = 1$ the agent ignores the immediate reward and considers all future ones. We set $\gamma = 0.99$.

The goal of this formalization is to train an agent that is able to find a policy π^* that *maximizes the expected cumulative reward*:

$$\pi^* = argmax_{\pi}\mathbb{E}[\sum_{t=0}^{|\pi|} \gamma^t r(s_t, a_t)] \tag{2}$$

where $|\pi|$ is the length of the policy which corresponds to the number of sessions for each user.

To implement our MDP, we propose to adapt SMORL, a state-of-the-art architecture for recommendation diversity [9]. An overview of the framework is displayed in Figure 1. It is split into three parts: **(A)** represents the summarising of the k previous items defining the last session. We used the same architecture as in [13] by replacing the LSTMs with GRUs. The session is embedded using this layer of GRUs followed by a layer of attention:

$$o_i = GRU(item_i|o_{i-1}),\ o'_i = ReLU(W_1 o_i + b_1)$$

$$S = \sum_{i=0}^{k} o'_i \frac{e^{W_2 o'_i + b_2}}{\sum_{j=0}^{k} e^{W_2 o'_j + b_2}}$$

(B) represents the adaptation of SMORL model [9]. SMORL is composed of two heads, the self-supervised and the RL regularizer. The former head is a fully connected layer that plays the role of a traditional recommender. It ranks all items by predicting their utility and is trained using a cross-entropy loss. The latter is an RL head which modifies the initial ranking of items as they are trained simultaneously. It learns the Q-function using the Scalarized Deep Q-learning (SDQL) [14] which is an extension of DQN [15]. The part **(C)** has the goal of learning the right attribute that maximizes diversity. It learns the Q-function using DQN.

V. Experiments

We run an extensive set of experiments to study the impact of our solutions on diversity as well as relevance and response time. Our code as well as our dataset are available on our GitHub repository[1].

A. Setup

We split the data in a user-wise fashion, i.e., for every user, we chronologically build sessions where each session contains k items. We use the l last sessions, of each user, as test set and the remaining ones as training set. To compare our algorithms with ones that do not capture attribute diversity, we implement MMR_G and SWAP_G that estimate the diversity in a global fashion.
Dataset. We use a real-world dataset, a merge between MovieLens 10M and IMDb datasets, which contains 5 real attributes: Genre, Duration, Release Year, Rating, and Type of movie. We generate a semi-synthetic dataset by augmenting the real-world one with $\{10, 20, 30, 95\}$ simulated attributes. The generation of these independent attributes is performed using different distributions: Gaussian, Exponential, Gamma, Uniform, and Zipfian. The choice of the distribution as well as its parameters, to generate a single attribute, is random. The generation is performed once at the beginning of the process. This dataset is sparse as 98.5% of data is missing.
Measures. We use T_u to denote the real session of user u and $R_u@k$ the session of k items recommended to u.

For each user u, precision measures the proportion of recommended items that are relevant:

$$Precision_u@k = \frac{|R_u@k \cap T_u|}{k}$$

[1] https://github.com/SessionDiversity/Multi-session-Diversity-Attributes

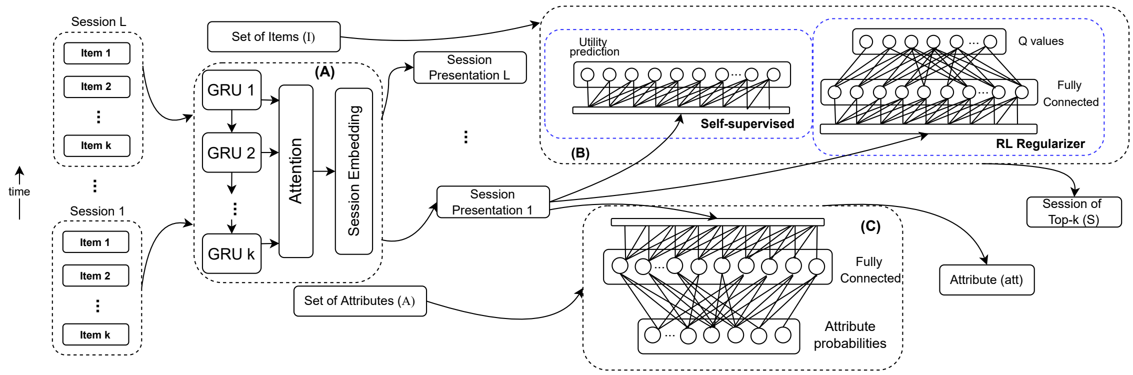

Fig. 1: Overview of the architecture of the RL framework. **(A)** represents the summarizing of a session into a latent space, **(B)** represents the SMORL model [9] to choose the next session items, and **(C)** designates the model for choosing the best attribute to optimize diversity.

An item is considered to be relevant to a user if it appears in the test set. We also use a diversity metric to quantify how diverse the recommended session is. For each user u:

$$Diversity_u = \frac{\sum_{i \in R_u@k} \sum_{j \in R_u@k \setminus \{i\}} div(i,j)}{k(k-1)}$$

where $div(i,j) = \frac{1}{1+cosine\_similarity(i,j)}$

We use two different multi-aspect metrics to measure the combination between accuracy and diversity. The first one is an adaptation of the F1-Score:

$$FScore_u@k = \frac{2.Precision_u@k.Diversity_u}{Precision_u@k+Diversity_u}$$

The second one is an adaptation of $nDCG$ [39] called $\alpha nDCG$. It has a tuning parameter $\alpha \in [0,1]$, which indicates the strength of penalization on the appearance of similar items in the recommended session. In the case where $\alpha = 0$, $\alpha nDCG$ is equivalent to $nDCG$. Given the session $R_u@k$:

$$\alpha nDCG = \frac{\sum ng(r)/log(r+1)}{\sum ng^*(r)/log(r+1)}$$

where $ng(r) = I_u(r)(1-\alpha)^{C_u(r-1)}$ which represents the novelty-biased gain at rank r. $I_u(r)$ is the relevance of the item at the rank r and $C(r) = \sum_{i=1}^{r} I_u(i)$.

Finally, we calculate for each user the response time needed to generate the recommended sessions.

Default values and Parameter tuning. We report results in the case where session size $k = 5$. They are aggregations of 3 runs over sets of sampled users. We performed a grid search over a set of parameters to fine-tune MMR and RL methods and find those yielding the best results. We provide details of all parameters in Table I and highlight the best ones. The parameter "Session Size" represents the size of the session embedding.

Methods	Parameters	Values
SMORL	Batch Size	256, **64**
	Session Size	**100**, 200, 500
	Learning Rate	0.0001, 0.0003, 0.001, **0.003**
MMR	Trade-off of relevance and diversity (α)	0.3, 0.5, **0.6**, 0.9

TABLE I: Parameter tuning values

B. *Summary of Results*

We found that our baselines are the best performers in terms of diversity regardless of the number of attributes or the number of sessions while SMORL has the best trade-off between diversity and accuracy. The RL variant is also the best time performer, especially for a large number of attributes.

We also found that SWAP_G is best in terms of response time as it does not iterate through attributes while MMR_G is outperformed by SMORL. SWAP_G and MMR_G are outperformed in terms of diversity and accuracy.

During this work, we used different RL architectures, simple DQN [15] and A3C [40]. We do not report their results as they are often outperformed by the SMORL adaptation and the baselines.

C. *Single Session Recommendation*

1) Diversity: Figure 2 reports the evolution of intra-diversity as a function of the number of attributes. The first observation is that the diversities of the baselines are better than SMORL's with a clear overall advantage for

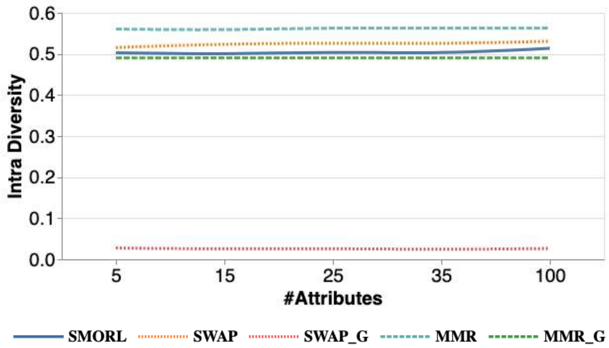

Fig. 2: Evolution of Diversity as a function of the number of attributes

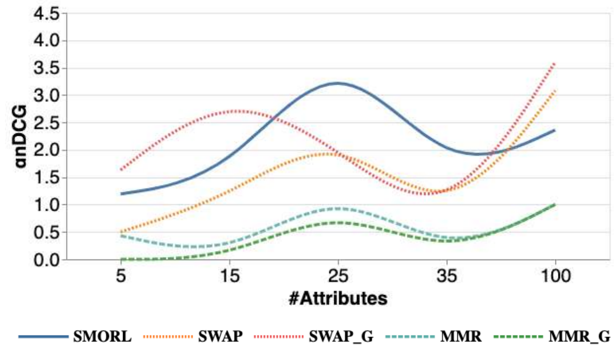

Fig. 4: Evolution of $\alpha nDCG$ as a function of the number of attributes

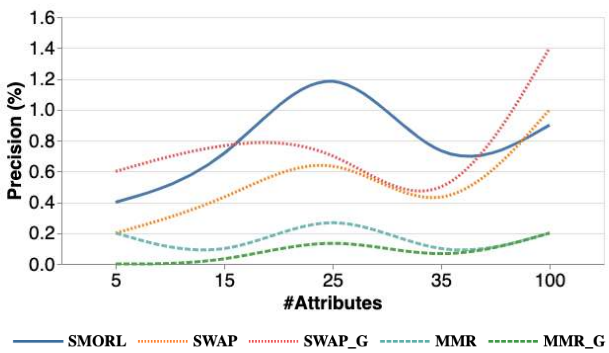

Fig. 3: Evolution of Precision as a function of the number of attributes

MMR. One possible explanation of the performances of MMR and SWAP is that they calculate the diversity for all attributes and choose always the best one. SMORL on the other hand relies on predicting diversity attributes which makes it more vulnerable and returns more false positives and that affects negatively its performance. For example, for $\#Attributes = 5$, SMORL has a precision of 25% for choosing the best attributes while SWAP and MMR have a precision of 100%.

The second observation is that the diversity of MMR_G and SWAP_G is smaller than others regardless of the number of attributes. This is because the latter methods tend to select the attributes that maximize diversity while no such optimization is performed for both MMR_G and SWAP_G

2) Accuracy: Figure 3 represents the evolution of precision as a function of the number of attributes. From the figure, we can see that, generally, the adaptation of SMORL is the best and outperforms the baselines (MMR and SWAP). As explained in the original paper [9], SMORL can identify users having diverse interests and recommend them suitable items. One possible explanation is that SMORL incorporates a traditional recommender. Indeed, the self-supervised head of SMORL plays that recommendation role and learns the most accurate next items to recommend in a sequential way.

We notice that SWAP performs better than MMR. The reason is that SWAP chooses at each step the best item to swap within the initial list of items while MMR chooses the next item using a linear trade-off function between utility and diversity. We also notice that the two variants of algorithms achieve a similar precision with an advantage of MMR over MMR_G and SWAP_G over SWAP.

3) Accuracy-Diversity: Figure 4 shows the evolution of $\alpha nDCG$ as a function of the number of attributes. One can see that SMORL is the best performer compared to SWAP and MMR. We can explain that by the fact that the RL head of SMORL is used to introduce more diverse items while the other head provides more accurate ones. The combination between these heads and their mutual learning permit the model to obtain a good balance between diversity and accuracy. Despite MMR having a better and increasing diversity, it is outperformed by SWAP regardless of the number of attributes. Indeed, the reason is, that this latter achieves a far better accuracy which results in having a better balance.

4) Time: Figure 5 reports the evolution of response time as a function of the number of attributes. We see that the baselines have the worst time performance

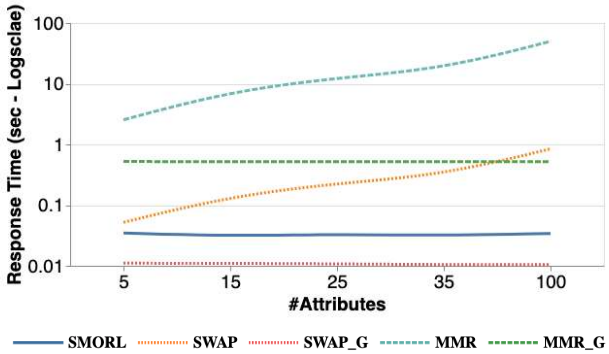

Fig. 5: Evolution of response time as a function of the number of attributes

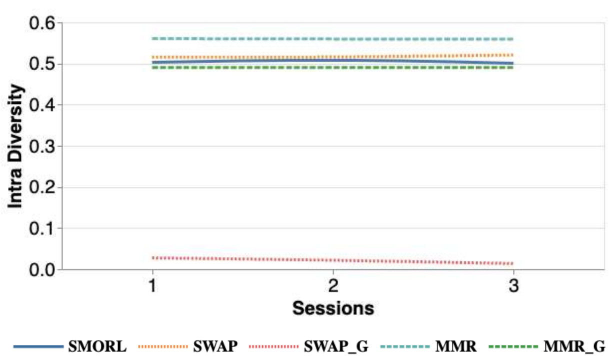

Fig. 6: Evolution of Diversity across sessions for $\#Attributes = 5$

independently of the number of attributes, with better results for SWAP. Indeed, these methods iterate over all attributes to choose the best one. This obviously makes the computation expensive and increases with the increase of the number of attributes and items. The second observation is that the RL algorithm has a constant time evolution across the number of attributes. SWAP_G is obviously the best performer as it does not iterate over attributes while MMR_G outperforms SWAP for $\#Attributes = 100$ and MMR.

D. Multiple Session Recommendation

In this section, we fix the number of sessions to $l = 3$.

1) Diversity: Figure 6 shows the evolution of diversity across multiple sessions. We observe that the diversities of all models are mainly constant regardless of the session. The models are designed to optimize the

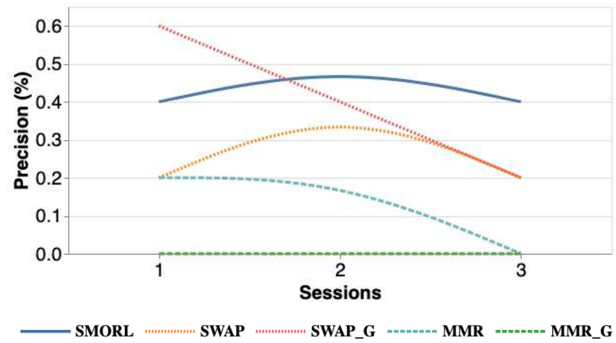

Fig. 7: Evolution of Precision across sessions for $\#Attributes = 5$

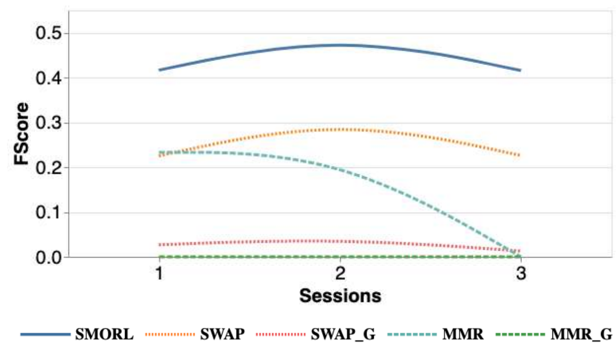

Fig. 8: Evolution of F-Score across sessions for $\#Attributes = 5$

diversity of a session and maintain its maximization for the next ones.

2) Accuracy: Figure 7 shows the evolution of precision across multiple sessions. The main observation is that the precision of SMORL and SWAP are both increasing then decreasing with an advantage for SMORL while MMR precision is continuously decreasing. The other observation is that MMR_G is the worst performer and SWAP_G is quickly decreasing and outperformed by SMORL.

3) Accuracy-Diversity: Figure 8 shows the evolution of F-Score across multiple sessions. We observe that despite the good results of MMR on diversity, the trade-off metric is quickly decreasing. SMORL and SWAP have the opposite behavior as they remain constant. We also observe that methods with attributes selection outperform SWAP_G and MMR_G.

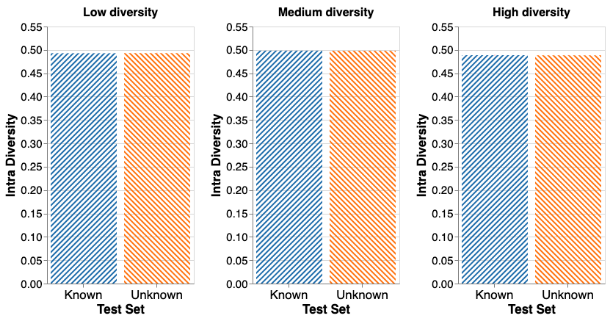

Fig. 9: Diversity of transfer learning for a single session

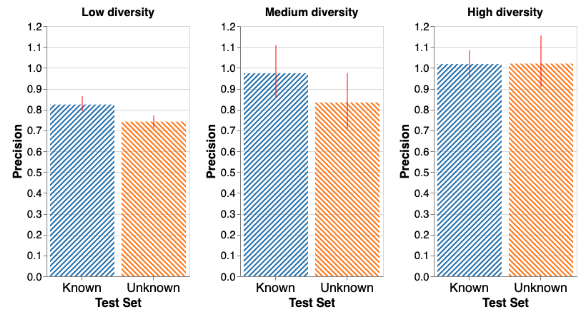

Fig. 10: Precision of transfer learning for a single session

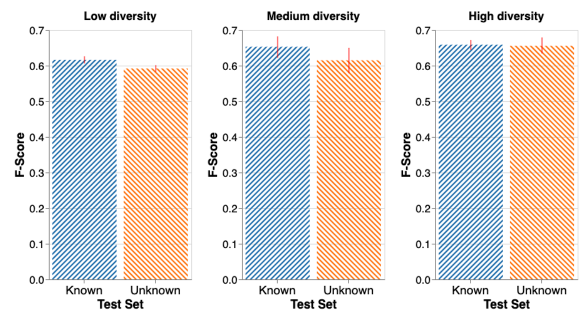

Fig. 11: F-Score of transfer learning for a single session

E. RL Transfer Learning

Our last experiment is about policy transfer. We split the users into three classes: users with high, medium, and low diversity using K-means [41]. The diversity of each user is the average diversities of the last 5 sessions in the training set. We split, user-wise, each class in a way that 75% of users, "known users", are used to train a model while the remaining 25% "unknown" ones are used to test the transfer of the model. We train for each class a SMORL model and test it on both types of users. The results are displayed in figures 9, 10, and 11.

From the figures, one can see that, overall, we can transfer SMORL models to users that were unknown to them. Indeed, in the "Low diversity" and "Medium diversity" cases, models do not maintain the level of precision and F-Score but the observed decrease is slight compared to "known" users. For example, we register a loss of 9% and 14% of precision for the "Low" and "Medium diversity" models respectively. In the case of "High diversity", precision and F-Score for "unknown" users are the same as for "known" ones. One can also observe that the results of diversity are the same between the test sets regardless of the type of diversity class.

VI. Conclusion

In this paper, we developed an approach for learning diversity attributes in multi-session recommendations. We implemented two different solutions: one based on traditional diversity algorithms and the other based on reinforcement learning. Our extensive experiments on a semi-synthetic dataset demonstrate that the RL-based approach is the best in terms of accuracy and response time, especially for a high number of attributes. In the future, we would like to extend our work to learn a diversity function that combines multiple attributes in each session.

References

[1] Q. Wu, Y. Liu, C. Miao, Y. Zhao, L. Guan, and H. Tang, "Recent advances in diversified recommendation," *arXiv preprint arXiv:1905.06589*, 2019.

[2] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proceedings of the 14th international conference on World Wide Web*, 2005, pp. 22–32.

[3] C. Yu, L. Lakshmanan, and S. Amer-Yahia, "It takes variety to make a world: diversification in recommender systems," in *Proceedings of the 12th international conference on extending database technology: Advances in database technology*, 2009, pp. 368–378.

[4] S. Vargas, L. Baltrunas, A. Karatzoglou, and P. Castells, "Coverage, redundancy and size-awareness in genre diversity for recommender systems," in *Proceedings of the 8th ACM Conference on Recommender systems*, 2014, pp. 209–216.

[5] L. Qin and X. Zhu, "Promoting diversity in recommendation by entropy regularizer," in *Twenty-Third International Joint Conference on Artificial Intelligence*. Citeseer, 2013.

[6] S. A. Puthiya Parambath, N. Usunier, and Y. Grandvalet, "A coverage-based approach to recommendation diversity on similarity graph," in *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016, pp. 15–22.

[7] M. Esfandiari, R. M. Borromeo, S. Nikookar, P. Sakharkar, S. Amer-Yahia, and S. B. Roy, "Multi-session diversity to improve user satisfaction in web applications," in *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, 2021, pp. 1928–1936.

[8] Y. Liu, Y. Zhang, Q. Wu, C. Miao, L. Cui, B. Zhao, Y. Zhao, and L. Guan, "Diversity-promoting deep reinforcement learning for

interactive recommendation," *arXiv preprint arXiv:1903.07826*, 2019.

[9] D. Stamenkovic, A. Karatzoglou, I. Arapakis, X. Xin, and K. Katevas, "Choosing the best of both worlds: Diverse and novel recommendations through multi-objective reinforcement learning," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 957–965.

[10] A. Anderson, L. Maystre, I. Anderson, R. Mehrotra, and M. Lalmas, "Algorithmic effects on the diversity of consumption on spotify," in *Proceedings of The Web Conference 2020*, 2020, pp. 2155–2165.

[11] J. Carbonell and J. Goldstein, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998, pp. 335–336.

[12] P. Fraternali, D. Martinenghi, and M. Tagliasacchi, "Top-k bounded diversification," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, 2012, pp. 421–432.

[13] C. Hansen, R. Mehrotra, C. Hansen, B. Brost, L. Maystre, and M. Lalmas, "Shifting consumption towards diverse content on music streaming platforms," in *Proceedings of the 14th ACM international conference on web search and data mining*, 2021, pp. 238–246.

[14] H. Mossalam, Y. M. Assael, D. M. Roijers, and S. Whiteson, "Multi-objective deep reinforcement learning," *arXiv preprint arXiv:1610.02707*, 2016.

[15] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[16] S. Wang, L. Hu, Y. Wang, L. Cao, Q. Z. Sheng, and M. A. Orgun, "Sequential recommender systems: Challenges, progress and prospects," *CoRR*, vol. abs/2001.04830, 2020.

[17] M. Quadrana, D. Jannach, and P. Cremonesi, "Tutorial: Sequence-aware recommender systems," in *WWW*, S. Amer-Yahia, M. Mahdian, A. Goel, G. Houben, K. Lerman, J. J. McAuley, R. Baeza-Yates, and L. Zia, Eds. ACM, 2019, p. 1316.

[18] H. Fang, D. Zhang, Y. Shu, and G. Guo, "Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations," *ACM Trans. Inf. Syst.*, vol. 39, no. 1, pp. 10:1–10:42, 2020.

[19] S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. A. Orgun, and D. Lian, "A survey on session-based recommender systems," *ACM Comput. Surv.*, vol. 54, no. 7, pp. 154:1–154:38, 2022.

[20] M. Ludewig and D. Jannach, "Evaluation of session-based recommendation algorithms," *User Modeling and User-Adapted Interaction*, vol. 28, no. 4, pp. 331–390, 2018.

[21] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, "Effective personalization based on association rule discovery from web usage data," in *Proceedings of the 3rd international workshop on Web information and data management*, 2001, pp. 9–15.

[22] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 811–820.

[23] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," in *Twenty-Third international joint conference on Artificial Intelligence*, 2013.

[24] B. Hidasi and A. Karatzoglou, "Recurrent neural networks with top-k gains for session-based recommendations," in *Proceedings of the 27th ACM international conference on information and knowledge management*, 2018, pp. 843–852.

[25] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *Proceedings of the eleventh ACM international conference on web search and data mining*, 2018, pp. 565–573.

[26] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li, "Drn: A deep reinforcement learning framework for news recommendation," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 167–176.

[27] X. Zhao, L. Zhang, Z. Ding, L. Xia, J. Tang, and D. Yin, "Recommendations with negative feedback via pairwise deep reinforcement learning," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1040–1048.

[28] X. Zhao, L. Xia, L. Zhang, Z. Ding, D. Yin, and J. Tang, "Deep reinforcement learning for page-wise recommendations," in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018, pp. 95–103.

[29] F. Pan, Q. Cai, P. Tang, F. Zhuang, and Q. He, "Policy gradients for contextual recommendations," in *The World Wide Web Conference*, 2019, pp. 1421–1431.

[30] M. Chen, A. Beutel, P. Covington, S. Jain, F. Belletti, and E. H. Chi, "Top-k off-policy correction for a reinforce recommender system," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 2019, pp. 456–464.

[31] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3, pp. 229–256, 1992.

[32] X. Chen, S. Li, H. Li, S. Jiang, Y. Qi, and L. Song, "Generative adversarial user model for reinforcement learning based recommendation system," in *International Conference on Machine Learning*. PMLR, 2019, pp. 1052–1061.

[33] X. Bai, J. Guan, and H. Wang, "A model-based reinforcement learning with adversarial training for online recommendation," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[34] X. Xin, A. Karatzoglou, I. Arapakis, and J. M. Jose, "Self-supervised reinforcement learning for recommender systems," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 931–940.

[35] Y. Liu, Z. Shen, Y. Zhang, and L. Cui, "Diversity-promoting deep reinforcement learning for interactive recommendation," in *5th International Conference on Crowd Science and Engineering*, 2021, pp. 132–139.

[36] T. Di Noia, V. C. Ostuni, J. Rosati, P. Tomeo, and E. Di Sciascio, "An analysis of users' propensity toward diversity in recommendations," in *Proceedings of the 8th ACM Conference on Recommender Systems*, 2014, pp. 285–288.

[37] T. Di Noia, J. Rosati, P. Tomeo, and E. Di Sciascio, "Adaptive multi-attribute diversity for recommender systems," *Information Sciences*, vol. 382, pp. 234–253, 2017.

[38] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[39] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon, "Novelty and diversity in information retrieval evaluation," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008, pp. 659–666.

[40] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, 2016, pp. 1928–1937.

[41] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern recognition*, vol. 36, no. 2, pp. 451–461, 2003.