REINFORCEMENT LEARNING RECOMMENDER SYSTEM FOR MODELLING LISTENING SESSIONS

Tomas Gajarsky
Moodagent
tg@moodagent.com

EXTENDED ABSTRACT

The superior quality of listening experience provided by a music service could convince the potential users to become customers. A reinforcement learning (RL) recommender system for generating playlists, which continuously learns from users' implicit feedback to maximize their satisfaction, was developed and compared to an established sequence-aware system and two traditional similarity-based methods. The first task was to place the tracks that are more relevant to the user higher in the playlist, and the second one was to re-create ordering patterns of the relevant tracks from the original listening sessions. A listening session is a set of tracks that the user has listened to in a particular order within a certain time frame. A track is relevant when it was played for more than half of its duration.

The main characters of RL are the agent and the environment. The environment is the world that the agent lives in and interacts with. At every interaction step, the agent observes the state of the environment and decides on an action to take, namely on a track to recommend. The state is described by tracks from the current listening session, the current user, and the current time. Tracks are represented by audio content features, as well as popularity features. The user is represented by a user taste vector, which is created by combining representations of relevant tracks from the user's listening history. Before the environment moves onto another step, the agent receives a reward, which is a signal that tells it how good or bad the decision was. A relative listening time is used as the reward here. The architecture of the RL agent, inspired by [1, 4], consists of the actor and critic networks. The actor first generates a representation of an ideal track (proto action) from the state. Then, the closest valid action to the proto action has to be found amongst the representations of real tracks using cosine similarity. Finally, the critic outputs an expected reward value which assesses the quality of the valid action-state pair. Both, the actor and the critic, consist of a state encoder followed by two hidden layers. Part of the state encoder is the session encoder, which consists of a recurrent neural network with attention mechanism that encodes the temporal dependencies between tracks in a sequence.

The framework is trained to follow the off-policy using Deep Deterministic Policy Gradient [3] in an offline scenario. Contrary to an online setting, the agent learns from a simulation where the valid actions are already given in the historical data. Thus, the gap between the valid action and the generated proto action is minimized during the training to connect the actor and critic. The actor's parameters are updated in the direction of maximizing the expected reward using the current state and proto action. The critic's parameters are updated in the direction of minimizing the difference between the expected reward and the immediate reward plus the future expected reward. In the testing scenario, the next track is selected by mapping the proto action to a number of similar valid actions, which are then judged by the critic. The valid action with the highest expected reward is selected to be the next track in the session.

The data used in this experiment is of a proprietary nature. It contains listening histories with corresponding user IDs, track IDs, timestamps, and amounts of time tracks were played for. Around $40,000$ individual listening sessions with an average number of 7 listening events per session were isolated from more than $480,000$ listening logs. In order to preserve time consistency, the data was split into training and testing subsets in a time-based fashion with ratio of 4 to 1.

Since the ground truth rewards of other tracks from different sessions are not transferable to the current session, the only way to test the system in an offline scenario is to evaluate its ability to reorder the listening events in the original sessions. The evaluation baselines are given in (a) the original data, (b) by randomizing the order of the listening events in the original sessions, and (c) by constructing ideal listening session that are ordered by relative listening time from the highest to the lowest, while preserving the order of relevant tracks. The examined methods are: (d) the similar to seed method, where the tracks are ordered according to their similarity with seed track of a given playlist; (e) the content-based user model, where the tracks are ordered according to the user vectors that represent a global taste of the individual users; (f) the GRU4REC [2] model, which is an established sequential method; and (g) our RL agent.

The most important metrics that quantify the quality of listening sessions in this project are the normalized discounted cumulative gain (NDCG), which is larger when more relevant tracks are placed higher in the playlist and the mean squared error (MSE), which is lower the closer ranks of tracks in the original sessions and in the reconstructed sessions are.

METHOD	NDCG (mean)	MSE (mean)
a) Original	0.8494	0.0
b) Randomized	0.8439	17.186
c) Ideal	1.000	7.366
d) Similar to seed	**0.8542**	15.874
e) Content-based user model	0.8507	16.408
f) GRU4REC	0.8503	**15.041**
g) RL agent	0.8517	16.382

Table 1. Methods comparison - results

The average results of all test listening sessions presented in Table 1 were close, therefore Wilcoxon signed rank tests were executed for the purpose of the final comparison. The following statistical significances were revealed. The RL recommender system was capable of placing tracks relevant to users higher than they were placed originally. On the other hand, it's ability to recreate ordering patterns of tracks is comparable to randomization. The GRU4REC model achieved opposite results, when it showed superior performance in the second task among the examined methods, but did not have enough power to adjust the recommendations for a specific user or context. The simplest method, which ordered the tracks according to their similarity to the seed track, proved to be efficient for both of the tasks in a given setup, despite the fact that it does not make use of any sequential or user information. However, this simple approach is not going to get any better, as opposed to the RL agent, which has a potential of dynamically learning about various strategies that may satisfy other user types in different situations in an online scenario.

ACKNOWLEDGMENTS

I thank my colleagues at Moodagent and professors at Aalborg University for the provided help and support.

REFERENCES

[1] Gabriel Dulac-Arnold, Richard Evans, Hado van Hasselt, Peter Sunehag, Timothy Lillicrap, Jonathan Hunt, Timothy Mann, Theophane Weber, Thomas Degris, and Ben Coppin. Deep reinforcement learning in large discrete action spaces. *arXiv preprint arXiv:1512.07679*, 2015.

[2] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[3] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[4] Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. Deep reinforcement learning for page-wise recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 95–103. ACM, 2018.