Music Recommendation Based On Listening Sessions

Li-An Yang A53242336
CSE 293 WI19

Abstract— **Nowadays, music streaming platforms account for 75% of the music industry revenue, with Spotify retaining 83 million premium account users worldwide. The market is still expanding as more personalized services come into play. In this report, we consider a new scheme for recommendation by separating listening histories into sessions for each user. Our goal is to attempt *next-item recommendation* in the latest listening session to examine the pros and cons of such arrangement.**

I. INTRODUCTION

The task of recommending the right piece of music at the right time has become increasingly important with the rise of streaming music industry. Under different situation, there exists multiple ways to provide music recommendation properly. For instance, Pandora has been working on automatic playlist generation [1] to improve their next-station prediction algorithms using recurrent neural network. The RecSys Challenge in 2018 sponsored by Spotify called for solution to the task of automatic playlist continuation. The goal is to generate tracks to extend user's current music taste and thereby 'continue' the playlist. Here we suggest another scheme based on user listening sessions. While playlists are maintained by users, listening sessions can be segregated by prolonged music pause between plays. We also assume that the first few tracks of a session should take more weights than those in the latter part in terms of recommendation performance.

The rest of this manuscript is ordered as follows. In section II, we investigate some state-of-the-art work in the field of recommeder systems. In section III, we show some experiments on top of the proposed scheme. We then move on to discuss about potential development and limitations in section IV. Lastly, we draw conclusion in section V.

II. RELATED WORK

There has been a sequence of development in recommending the right pieces of music. We first mention the classical latent factor model and a few of its variations. Next, sequence-aware models are introduced by exploiting Markov Chain and Markov Decision Process. Lastly, we involve deep neural networks and discuss about some works using RNN to deliver state-of-the-art results.

A. Latent Factor Models for Collaborative Filtering

General model in a recommender system considers each user's interaction with the item instead of features from both side separately. The user-item matrix, constructed from explicit or implicit feedback, is then factorized into latent factors to represent user's (item's) preference (characteristic). Unobserved interactions can therefore be estimated by the inner product of user and item embedding vectors.

t

Statistics	Lastfm-1k	MLHD
Number of users	971	1054
Number of unique tracks	1,500,659	248,885
Number of unique artists	177,023	77,545
Avg. number of entries per user	9,305	28,565
Avg. play count per track	12	121
% of tracks that only appear once	35%	15.7%
Time span	2005-02 2009-04	2005-02 2013-09

TABLE I
STATSTICS OF OUR DATASET

As opposed to optimizing mean-squared error, Steffen et al. [2] proposed a pair-wise method that optimizes a user u's preference of item i over j. The contribution is significant in the situation where the dimension of user or item is high. Particularly, it is a ranking loss that converges much faster than point-wise loss.

B. Sequence-aware Recommendation

Several methods focus on modeling the sequential pattern of user behavior as well. McFee et al. [3] considered playlists as strings of songs in the realm of natural language processing. The problem can therefore be tackled with Markov Chain to model the distribution of a naturally occurring playlist. Based on this work, Chen et al. further embedded songs into latent space before integrating them in a Markov Chain sequence model.

C. Recurrent Neural Network

Due to the ability to capture nonlinear relationship, RNN has been adopted by several works to tackle sequential recommendation problems [4]. Massimo et al. [1] combined Gated Recurrent Unit (GRU) and a Learning-to-Rank model to make next-station predictions for Pandora. On the other hand, Wang-Cheng et al. [5] apply attention mechanism on top of the RNN framework to consider not only long-term dependencies but also more recent activities. Deep learning-based recommender systems have been shown to outperform classical models while still carrying great potential for further improvement.

III. EXPERIMENTS

In this section, we explain the procedure of our experiments with empirical results. Our goal is to verify the feasibility of our session-based scheme with several classic models.

Dataset	Metric	PopRec	BPR-MF	BPR-MF@20	BPR-MF@10	FPMC	FPMC@20	FPMC@10	SASRec
Lastfm-1k	Hit@10	0.661	0.379	0.433	0.42	0.384	0.406	0.436	0.445
	NDCG@10	0.463	0.246	0.28	0.274	0.254	0.255	0.282	0.326
MLHD	Hit@10	0.625	0.408	0.475	0.489	0.4	0.479	0.491	0.604
	NDCG@10	0.4	0.233	0.303	0.309	0.234	0.298	0.316	0.412

TABLE II

RECOMMENDATION PERFORMANCE FOR NEXT-TRACK RECOMMENDATION

A. Datasets

We evaluate our methods on two datasets featuring listening history of around 1K users. Each row typically consists of artist and track ID as well as the timestamp associated with the play event.

- **Lastfm-1k** As the name suggests, the dataset consists of a large collection of listening entries for up to 1k users. The data was extracted from Last.fm using its API. There are on average 19,305 entries of history for each user with 35% of all tracks that have only one appearance. It can be inferred that there might exist a cold-start problem due to sparsity of the dataset. The distribution of play counts for each track is plotted in Figure 1, following power law.
- **The Million Listening History Dataset (MLHD)** MLHD is a large-scale collection of music listening events extracted from Last.fm. It is orders of magnitude larger than other music listening history dataset. On average, it has 28,565 valid logs for each user with 15.7% of all tracks that have appeared just once. We can say it is a more dense dataset compared to Lastfm-1k within a similar time span. For testing purposes and due to computational limitations, we have only used 1GB out of 576GB of the data.
  Besides listening histories, it also provides a set of user profiling features, of which we didn't make use in our experiment. The statistics of both datasets are shown in Table I.

B. Methods

Next, We list several classic methods adopted for our experiments.

- **PopRec**: This is a simple model which makes prediction solely based on the popularity of track.
- Bayesian Personalized Ranking (BPR) [2]: BPR is based on the optimization of a pair-wise ranking loss function using traditional methods such as matrix factorization from implicit feedback.

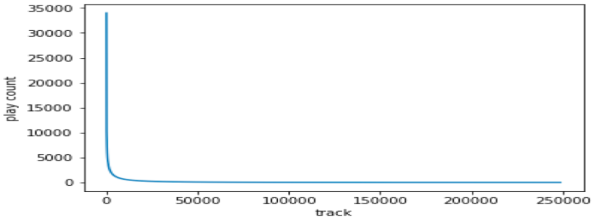

Fig. 1. Distribution of the play counts for Lastfm-1k.

- **Factorized Personalized Markov Chains (FPMC)** [6]: FPMC incorporates maximum likelihood estimation in Markov Chains and matrix factorization methods to capture long-term item-to-item transitions.
- **SASRec**: A state-of-the-art work in sequential recommendation which focuses on striking the balance between self-attention and sequential neural networks.

C. Procedures

As aforementioned, we separate listening entries into sessions, which are defined as a list of play events without obvious interruption. We therefore define a gap to decide whether or not the current play event should trigger a new listening session or be included in the current session. Out of testing purposes, we set the value of gap at 8,000 seconds and continue our procedure.

Our assumption is that it is more reasonable to model the first few tracks of a session since these tracks should be more representative of user's music taste. As music listening is considered a passive behavior [7], latter tracks in a session are usually among those provided or recommended by the music platform, which in reality might not reflect the preference as much.

The procedures are listed as follows.

1) Remove noisy data having incomplete IDs.
2) Segment each user's listening history into multiple valid sessions. We denote $|S|$ as the number of sessions.
3) We consider the first track in the $|S|-1_{th}$ session and the $|S|_{th}$ session as validation data and test data for each user, respectively.
4) Train models on the first $|S|-2$ sessions.
5) Optimize our models on the validation set and verify on the test set.

D. Evaluation Metrics

We adopt two common evaluation metrics, Hit@10 and NDCG@10. Hit@10 represents the percentage of the ground-truth tracks that are among our top-10 recommendations. NDCG is measured by the position of relevant items and hence would be higher when the ground-truth track appears earlier in our ranked list of recommendation.

Along with the first track of the session as ground-truth, we randomly sample 100 negative tracks and rank these 101 items together, as in [8]. Recommendation performance is

Dataset	Metric	PopRec	BPR-MF	BPR-MF@20	BPR-MF@10	FPMC	FPMC@20	FPMC@10	SASRec
Lastfm-1k	Hit@10	0.522	0.294	0.346	0.349	0.294	0.333	0.336	0.334
	NDCG@10	0.354	0.179	0.211	0.211	0.181	0.211	0.208	0.241
MLHD	Hit@10	0.472	0.305	0.369	0.373	0.312	0.371	0.386	0.529
	NDCG@10	0.276	0.164	0.191	0.204	0.167	0.201	0.218	0.361

TABLE III

RECOMMENDATION PERFORMANCE FOR NEXT-NEW-TRACK RECOMMENDATION

measured based on the rankings before averaging the value of each user up to get the overall Hit@10 and NDCG@10.

E. *Tasks and Results*

Here we divide our problem into two slightly different tasks for different applications, *next-track prediction* and *next-new-track prediction*. The former can be applied to users at the start of any session(play events), while the latter strives to enhance discoverability for users, thereby recommending new tracks that might be of the user's preference.

The empirical results for both tasks are shown in Table II and III. From the comparison between models, we can observe that user preference is generally dominated by song popularity. Classic models like BPR-MF and FPMC fail to reach the same performance as PopRec by a considerable margin. Ranking the tracks by their popularity still stands the test for first-track recommendation in a new session. On the other hand, SASRec is located somewhere in the middle. Its NDCG@10 on MLHD for the first task even outperforms PopRec. Due to its ability to capture non-linear relations, it is expected that SASRec can produce better results than traditional methods, even by looking at only the last 50 items by default.

Also, it is also indicated that there exists a large gap in both Hit@10 and NDCG@10 between next-track and next-new-track recommendation. The former next-item task outperforms the latter by 8% to 15%. It is actually straightforward to see that predicting the next preferable new track is more challenging since users are more likely to favor a familiar song; therefore making it easier for our top-10 recommendation to hit the specific track.

IV. DISCUSSION

In this section, we discuss about a few implications out of our experiment. We also summarize from our discoveries whether or not the proposed session-based recommendation scheme retains further research values.

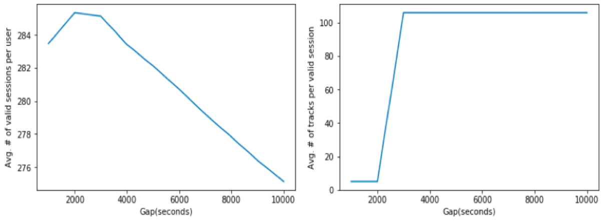

Fig. 2. (Left) Avg. # of valid sessions, (Right) Avg. # of tracks per valid session v.s. Gap seconds

A. *Session-wise prototype selection*

We have stated in our assumption that the first few tracks in a session should potentially carry more weights in next-item recommendation. To verify, we adopt a session-wise prototype selection by including only the first K tracks (K=10, 20) of each session to our training set. The results aligned with those using the original method are shown in Table II and Table III, marked as *BPR-MF@K* and *FPMC@K*. It is clear that prototype selection improves Hit@10 and NDCG@10 with only a subset of training set. For example, *BPR-MF@10* selects 64% of data to achieve a relative improvement of 20%. In general, a good model should be able to scale with the volume of data. Hence, in our case we can say traditional methods like BPR-MF and FPMC are not as suitable in session-based recommendation.

B. *The effect of data sparsity*

Next, we examine the difference in performance between the two datasets. We can observe that it is more difficult to make predictions on Lastfm-1k then on MLHD. It is reasonable because Lastfm-1k is said to be a lot more sparse than MLHD, with more than 20% of tracks that appear only once. Such dataset wouldn't work well with our interaction-based models. From Table I, it is obvious that MLHD contains more frequent listening behaviors and more average appearance for each track than Lastfm-1k.

C. *Choice of gap seconds*

One hyper-parameter tuning that can be of great importance is the choice of gap seconds. We test on a few values of gap from 1000s(16min) to 10000s(2.8hr), and plot them against the average number of sessions and number of tracks per session in Figure 2. It is worthy of note that here we only collect valid sessions in which there are more than ten distinct tracks. To our surprise, the average number of sessions does not vary much. It also appears that the average number of tracks per session converges at 106 after raising the value of gap above 2,000 seconds.

D. *Characteristic of music listening*

Normally, users is active on a platform before leaving implicit or explicit feedback, e.g., watching video or browsing items on the website. Music listening is, however according to [7], regarded as a passive consumption of songs. In other words, not every play event is truly representative of user's music taste. It might simply come from a recommendation service provided by say, Spotify. This is part of the reasons why we try to include only the first few tracks of a session as our training set, as they are more likely the tracks that users have searched or clicked on.

Considering the above uncertainty, we conclude that currently such arrangement of separating listening histories into sessions would not be an appropriate scheme for music taste extraction and personalization. User-generated playlist, on the other hand, could serve as a critical type of entity segmentation since track addition to a playlist is a conscious move and a strong preference indicator. Consequently, how to exploit orders of track addition in the task of automatic playlist continuation will be our next step to examine.

E. limitations

Due to the lack of computational resource, we are only able to test on the dataset of around 1,000 users. The volume of data could scale up to 576,000 users for MLHD if with sufficient computation power, and our empirical results might be more persuasive and representative.

Another limitation arises from the cold-start problem of data due to sparse user-item interaction. To cope with, we can instead extract audio features for each track and complement with similarity-based methods, such as nearest neighborhood, in our recommendation algorithms.

V. Conclusion

In this manuscript, we have studied the feasibility of segmenting listening histories into sessions and provided recommendation accordingly. We further design two tasks: next-track prediction and next-new-track prediction to come up with more comprehensive evaluations. The experimental results imply that such setting still requires more particular specification and hence is not stable enough yet to stand the quality test of recommendation services. Throughout the process of this work, we have realized a much more thorough understanding about the unique characteristics of music recommendation. In the future, we plan to extend our experience to the scope of music playlist and work on automatic playlist generation and continuation.

References

[1] Quadrana, M., Reznakova, M., Ye, T., Schmidt, E., Vahabi, H. “Modeling Musical Taste Evolution with Recurrent Neural Networks”. In: *arXiv preprint arXiv:1806.06535.* (2017).

[2] Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L. “BPR: Bayesian personalized ranking from implicit feedback”. In: *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence (pp. 452-461)* (2009).

[3] McFee, Brian, and Gert RG Lanckriet. “The Natural Language of Playlists”. In: *ISMIR* (2011).

[4] Zhang, S., Yao, L., Sun, A., Tay, Y. “Deep learning based recommender system: A survey and new perspectives.” In: *CSUR* (2019).

[5] Kang, W. C., McAuley, J. “Self-Attentive Sequential Recommendation”. In: *ICDM* (2018).

[6] Rendle, S., Freudenthaler, C., Schmidt-Thieme, L. “Factorizing personalized markov chains for next-basket recommendation.” In: *WWW* (2010).

[7] Schedl, M., Zamani, H., Chen, C. W., Deldjoo, Y., Elahi, M. “Current challenges and visions in music recommender systems research.” In: *nternational Journal of Multimedia Information Retrieval* (2018).

[8] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua. “Neural collaborative filtering”. In: *WWW* (2017).