User-Centric Evaluation of Session-Based Recommendations for an Automated Radio Station

Malte Ludewig
TU Dortmund, Germany
malte.ludewig@tu-dortmund.de

Dietmar Jannach
University of Klagenfurt, Austria
dietmar.jannach@aau.at

ABSTRACT

The creation of an automated and virtually endless playlist given a start item is a common feature of modern media streaming services. When no past information about the user's preferences is available, the creation of such playlists can be done using session-based recommendation techniques. In this case, the recommendations only depend on the start item and the user's interactions in the current listening session, such as "liking" or skipping an item.

In recent years, various novel session-based techniques were proposed, often based on deep learning. The evaluation of such approaches is in most cases solely based on offline experimentation and abstract accuracy measures. However, such evaluations cannot inform us about the quality as perceived by users. To close this research gap, we have conducted a user study (N=250), where the participants interacted with an automated online radio station. Each treatment group received recommendations that were generated by one of five different algorithms. Our results show that comparably simple techniques led to quality perceptions that are similar or even better than when a complex deep learning mechanism or Spotify's recommendations are used. The simple mechanisms, however, often tend to recommend comparably popular tracks, which can lead to lower discovery effects.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Collaborative filtering**; **Music retrieval**.

KEYWORDS

Session-based Recommendation; Music Recommendation; Quality Perception

ACM Reference Format:
Malte Ludewig and Dietmar Jannach. 2019. User-Centric Evaluation of Session-Based Recommendations for an Automated Radio Station. In *Thirteenth ACM Conference on Recommender Systems (RecSys '19), September 16–20, 2019, Copenhagen, Denmark.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3298689.3347046

RecSys '19, September 16–20, 2019, Copenhagen, Denmark

ACM ISBN 978-1-4503-6243-6/19/09…$15.00
https://doi.org/10.1145/3298689.3347046

1 INTRODUCTION

Modern online media streaming services often provide the functionality of an automated *radio* or *playlist*, where users provide a start item (e.g., a song or a video), and the system then automatically plays a virtually endless list of related items. In some cases, for example, on YouTube or Spotify, users can also give feedback on the played items, e.g., by "liking" them, by explicitly expressing that they are not interested, or by skipping to the next item. In the best case, this feedback is immediately taken into account by the system, leading to an update of the items in the playing queue.

In case the system is used by a new or anonymous user, playlists can be generated with *session-based* recommendation techniques [23]. These approaches do not require long-term preference profiles but are able to make recommendations *solely* based on the most recent interactions of a user. Due to their practical importance also in other domains, e.g., in e-commerce settings, a number of algorithmic proposals for session-based recommendation were made over the years. Technically, different strategies can be applied from traditional association rule mining, over nearest neighbor methods, to recent deep learning based techniques [10, 15, 16].

In the research literature, the comparison of session-based recommendation techniques is mostly based on *offline* experimentation. A recent comparison of techniques in [18], to some surprise, indicated that comparably simple approaches often work at least as good as one of today's more complex state-of-the-art techniques in terms of measures like precision and recall. Given these results, the question arises if such simple techniques only perform well when using such measures or if they are actually able to generate recommendations that are also perceived to be of high quality by users.

To address this question, alongside an offline evaluation, we conducted a between-subjects user study (N=250), where participants interacted with an online radio application that was developed for the experiment. The participants could provide a start track and then received automated recommendations. We compared five algorithms, which included both simple and more complex ones, and the recommendations retrieved through Spotify's API. We analyzed both the participants' observed behavior (e.g., in terms of the number of liked or skipped tracks) as well as the answers they provided in a post-task questionnaire.

One main outcome of the study is that some simple methods are comparable or even better in terms of their quality perception than complex ones, which means simple methods are not only competitive in offline evaluations. However, the tested simpler methods, by design, often recommend more popular tracks, which might lead to fewer opportunities for item discovery. Beyond the specific aspects investigated in the study, our work therefore emphasizes the known need for multi-faceted evaluation approaches and the consideration of domain-specific aspects when comparing algorithms.

2 RELATED WORK

2.1 Next-Track Recommendation Algorithms

A simple technique to determine the next tracks to play is to count track co-occurrences in past sessions and apply a recommendation scheme of the form "Customers who bought … also bought …". Such an approach can be considered to be one of the simplest session-based techniques, which is widely used in practical applications and which, depending on the domain, can already lead to good performance [18]. Over the years, however, a variety of more elaborate next-item prediction techniques for different domains were proposed, from sequential pattern mining over Markov models, embedding-based approaches, and, most recently, to deep learning techniques [4, 8, 9, 15, 16, 21, 25].

In particular in the music domain, also application-specific techniques were applied. Recent research for example showed that recommending popular tracks from artists that are similar to those that the user currently listens to can be quite effective, both in terms of accuracy measures [3] and in terms of the users' quality perception [13]. Similarly, also quite simple nearest-neighbor techniques proved to be effective not only in the music domain, but also in others, e.g., in e-commerce [18]. However, such neighborhood-based methods can exhibit a bias to recommend mostly popular items. The authors of [19] therefore propose to adjust the ranking of items based on their popularity.

In our study, we will compare algorithms of different families. Specifically, we consider one deep learning technique, an artist-based collaborative method, a popularity-aware nearest-neighbor technique, as well as simple item co-occurrences. In addition, we consider the recommendations by a commercial service, in our case the recommendations provided through Spotify's API.

2.2 User Studies

Differently from recent user studies that are focusing on specific aspects of recommendations—such as item similarity [27], presentation aspects [26], the inaction of users [29] or the effects of playlist recommenders on users [12]—our goal is to assess the *quality perception* of different session-based recommendation algorithms in a realistic scenario.

The number of user studies on the perception of music recommendations is comparably low. The two most similar examples of past research are [1] and [13]. In [1], the authors compared a number of "playlisting" approaches through a user study. In their experiment, the participants evaluated entire playlists—two at a time—as produced by different algorithm for a given seed song. They included collaborative techniques, an artist-based one, as well as Apple's iGenius system. In [13], in contrast, the participants did not assess a set of generated playlists, but had to indicate the suitability of four alternative tracks as possible continuations for a given playlist. From a technical viewpoint, the authors considered two neighborhood-based techniques and an artist-based one when determining the possible continuations.

While these two studies share some similarities with our work, their focus and experiment designs were different. Instead of evaluating entire playlists or individual continuations, our goal is to assess the users' quality perceptions in a more interactive setting that is common on modern music streaming sites. Specifically, in our study, the participants can give immediate feedback to the currently played tracks and the system will then automatically update the recommendations accordingly. Similar to [1], we also include a commercial playlisting service in our study. And, like in [13], we contrast offline accuracy results with the users' quality perceptions.

Outside the music domain, various user studies on the quality perception of recommenders were made in the past, e.g., for the movie domain [6, 28]. Recent insights show that it is important that study participants actually "consume" the items they assess [11, 17], which is also the case in our study.

Generally, user studies often rely on user-centric evaluation frameworks as proposed in [14] and [22]. While we do not directly apply these general frameworks in our very specific setup, we partly based our questionnaire items on the considerations presented in these frameworks.

3 STUDY DESIGN

Our main research goal is to understand how different algorithms affect the quality perception of users. Therefore, we developed an own online radio station to conduct the corresponding study.

3.1 Tasks for Participants

(1) Before using the radio, the participants were informed about their tasks and the expected duration of the study. They were also asked to provide informed consent to the terms of the study.

(2) They were then directed to a search interface, where they could enter a query to find a start track. They could listen to excerpts of the retrieved tracks and select one of them to start the radio.

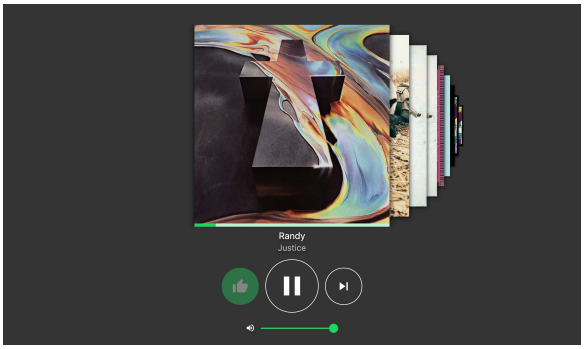

Figure 1: Radio Interface

(3) After this selection, the main radio application was started, see Figure 1. The radio then played the 30-seconds representative track excerpts provided by Spotify. Besides pausing the track and skipping to the next track, the participants could also use a "thumbs up" button to express that they like the current track. At all times, the radio interface provided a visual cue for the users that indicates that there is a list of upcoming tracks. This list was updated after a "thumbs up" or skip action, i.e., the participants received feedback that their actions had an effect.

For each played track, the participants were also asked to provide information if (i) they already knew the track, (ii) if the track matched the previously liked tracks, and (iii) if they liked the track in general, see Figure 2. Proceeding to the next track was only possible after the responses for each track were provided. Overall, each participant was asked to interact with and rate at least 15 tracks.

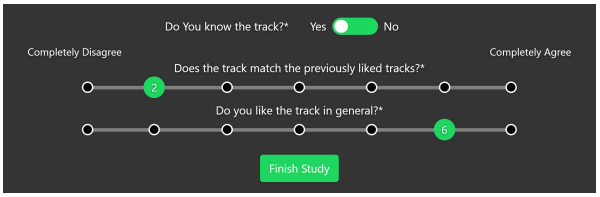

Figure 2: Rating Interface

(4) Once enough tracks were listened to, the participants could proceed to finish the study. In this final part, the participants were asked 11 questions (using 7-point Likert-type items) about their quality perceptions and intention to share or reuse the system. Table 1 shows these 11 questions, where the additional question Q8 is an attention check to assess if the participants answered the questionnaire with care. Furthermore, we asked four questions about their music enthusiasm (not shown in the table).

Table 1: Questions about Users' Quality Perceptions

	Question
Q1	I liked the automatically generated radio station.
Q2	The radio suited my general taste in music.
Q3	The tracks on the radio musically matched the track I selected in the beginning.
Q4	The radio was tailored to my preferences the more positive feedback I gave.
Q5	The radio was diversified in a good way.
Q6	The tracks on the radio surprised me.
Q7	I discovered some unknown tracks that I liked in the process.
Q8	I am participating in this study with care so I change this slider to two.
Q9	I would listen to the same radio station based on that track again.
Q10	I would use this system again, e.g., with a different first song.
Q11	I would recommend this radio station to a friend.
Q12	I would recommend this system to a friend.

3.2 Recommendation Strategies

The independent variable in our study is the assigned recommendation algorithm[1]. We employed five different strategies (see Table 2) and relied on Spotify's Million Playlist Dataset[2], abbreviated as MPD, as a basis for the recommendations.

Playing several tracks of the same artists within a short period of time is uncommon for automated radio stations. Since most tested methods are not designed to take this domain particularity into account, we designed a diversifying post-processing strategy, which we applied to all playlists returned by the algorithms. Specifically, we made sure that there are no artist repetitions within the next 3 tracks by re-ranking the tracks in the playlist. For the recommendations returned by Spotify's API, we furthermore removed tracks that did not appear in the MPD dataset. Thereby, we guaranteed a fair comparison of the techniques, in which all approaches recommend from the same catalog of items.

We optimized the hyper-parameters for the different strategies in an offline experiment on a subset of the MPD dataset. From a random sub-sample of 100,000 playlists we randomly selected 2,000 playlists as a validation set, while using the remaining playlists for model training. Parameter tuning was performed with a randomized search strategy in 100 iterations to find the best configuration for the mean reciprocal rank at list length 5.

[1]We used a round-robin assignment scheme as done in [13].
[2]https://recsys-challenge.spotify.com/

Table 2: Tested Recommendation Strategies

AR	A simple method based on association rules of length two, see [18].
CAGH	Recommends the greatest hits of artists similar to those liked in the current session [3]. The similarity is based on artist co-occurrences in user-provided playlists and approximated with matrix factorization.
GRU4REC	A recent session-based algorithm based on Recurrent Neural Networks (v2.0) [9]. The algorithm hyper-parameters were optimized for the mean reciprocal rank at list length 5.
S-KNN	A session-based nearest-neighbor approach proposed in [19] that lowers the predicted relevance scores for highly popular items. We used 500 as the number of neighbors, and set the sample size to 1000.
SPOTIFY	Recommendations in this treatment group were retrieved in real time from Spotify's API. Tracks that are not present in the MPD were excluded from the recommendations.

4 RESULTS

We have recruited 316 participants over Amazon's Mechanical Turk crowdsourcing platform ("Masters" only), ending up with reliable submissions from 250 unique users, i.e., 50 in each treatment group. The remaining submissions were eliminated as the users did not pass the attention check. On average, the participants needed about 15 minutes to complete the task. A majority of the users (nearly 80%) was from the US; over 50% of the participants were aged between 25 and 34.

On average and across all treatment groups, the participants listened to around 16 tracks (slightly above the minimum requirement), with an average pure listening time of 5.5 minutes. There were no significant differences in these respects.

Number of likes. The average number of *likes* per user were as follows: SPOTIFY: 4.48, GRU4REC: 5.36, CAGH: 5.38, S-KNN: 5.63, AR: 6.48. The AR method led to significantly[3] more likes than CAGH, SPOTIFY, and GRU4REC ($p<0.05$). Furthermore, the S-KNN method received 5.6 likes on average, which was significantly higher than SPOTIFY as well. All other differences were not significant.

Popularity. Looking at the average popularity of the recommended tracks[4], we found that SPOTIFY's and GRU4REC's recommendations were the least popular ones while AR and CAGH tend to recommend mostly popular items. Generally, the number of received likes per playlist highly correlates with the average track popularity of a list (r=0.89).

Individual Track Ratings. Table 3 shows our observations regarding the feedback for the individual tracks as shown in Figure 2. In terms of the percentage of already known tracks, we see that Spotify's recommendations are significantly less often known (or: more novel) than those of the other techniques.

When asking participants to what extent each track matches the previously liked ones, we observe that S-KNN and CAGH work best, while AR leads to recommendations that match the current playlist the least. The differences between S-KNN and the methods AR, GRU4REC, and SPOTIFY are significant ($p<0.05$).

Finally, looking at the average track "rating" in general, the differences between the algorithms are small. Somewhat surprisingly, the tracks produced by the AR method were the least liked ones, often with a significant difference, even though the average track

[3]Throughout the work, we use one-way ANOVA and a subsequent Tukey post-hoc test when the pre-requisites for these tests are fulfilled. Otherwise, we applied a Kruskal-Wallis test and a subsequent Mann-Whitney-U test.
[4]Computed based on the number of track occurrences in the MPD dataset.

popularity was the highest and they received the most like statements. This phenomenon can be explained when looking at the *mode* of the answers. The most frequent response (mode) was 1 for AR, whereas it was 7 for all other recommendation strategies. This means that AR probably recommended many "controversial" tracks that the users did not like even though they are very popular, e.g., because they matched previous tracks the least.

Table 3: Statistics for Item-Specific Questions (Mean and Standard Deviations)

Algorithm	Track known (%)	Track matches the playlist	Like the track in general
AR	8.61	4.06 ±1.60	4.34 ±1.37
CAGH	10.83	5.15 ±1.14	5.03 ±1.22
GRU4REC	9.30	4.61 ±1.52	4.94 ±1.31
S-KNN	10.13	5.31 ±1.04	4.94 ±1.06
SPOTIFY	7.00	4.72 ±1.13	4.69 ±1.07

Post-Task Questionnaire. For the Questions Q1 to Q12, we looked mostly at the median and mode values[5] and analyzed differences with the non-parametric tests.

For **Q1**, we found that the recommendations of S-KNN (median: 6)[6] were significantly more liked than those of AR, GRU4REC and SPOTIFY ($p<0.05$) and were perceived to be slightly better also than CAGH. All methods matched the user's general taste well (**Q2**), with median values of 5 (AR, SPOTIFY) or 6 (other methods). The values for AR were significantly lower than for CAGH, S-KNN, and GRU4REC.

The general pattern that AR performs worst and S-KNN best was also found for **Q3**, where S-KNN performed significantly better at finding tracks that match the seed track than AR and also GRU4REC. Regarding the adaptiveness of the radio upon user feedback (**Q4**), the median values were generally comparably high, ranging between 5 and 6. The analysis revealed that only AR performed significantly worse than CAGH, S-KNN, and SPOTIFY.

No differences between the group were found however regarding the diversification of the radio (**Q5**) and the surprise level (**Q6**). In terms of the discovery of unknown but liked tracks (**Q7**), SPOTIFY excelled. The responses were higher than for CAGH, S-KNN, GRU4REC, and AR ($p<0.1$). For the other methods, no significant differences could be found.

Regarding the last block of questions (**Q9** to **Q12**) about the users' intention to reuse the system or recommend it to friends, the values for S-KNN, CAGH and SPOTIFY are slightly higher than for the other techniques. The differences between these three methods and AR were always significant ($p<0.05$). For **Q10** and **Q12**, the responses for S-KNN and CAGH were also significantly higher than for GRU4REC.

4.1 Offline Accuracy vs. User Experience

The final goal of our work was to compare the users' quality perception with accuracy results obtained from offline experiments, since previous work [2, 5, 7, 20, 24] suggests that there can be a discrepancy. We therefore evaluated the different algorithms using five random sub-samples of the MPD dataset following the same procedure as for the parameter optimization described in Section 3.2. Furthermore, we applied the sequential evaluation protocol from [9]. In our work, we measure *precision* (P) and *recall* (R) in the usual way, comparing the list of recommended items with the next tracks in the playlist. In addition, we report the specific measurement method used in [9], which only considers the immediate next item, using the *hit rate* (HR) and the *mean reciprocal rank* (MRR). In order to design the evaluation as close as possible to the online application, we only proceeded through the first 15 entries of each test playlist to match the number of necessary interactions in the study.

Table 4: Offline Results

Algorithm	P@5	R@5	HR@5	MRR@5
S-KNN	**0.271**	**0.044**	0.137	0.077
GRU4REC	0.161	0.028	**0.151**	**0.096**
AR	0.234	0.037	0.135	0.081
CAGH	0.172	0.024	0.052	0.026
SPOTIFY	0.009	0.001	0.002	0.001

Table 4 shows that the ranking of the algorithms depends on the choice of the measurement method. The results indicate that the usual precision and recall measurement correlates better with our observations of the users' perception, with the S-KNN method leading to the best results in the offline experiment. The performance of SPOTIFY is very low in this comparison. In some ways, this is in line with the observations from the user study, which indicates that SPOTIFY's algorithm seems to be more optimized for discovery than for precision or recall on historical data.

5 SUMMARY OF FINDINGS

We have conducted a user study that investigates the quality perception of adaptive music recommendations in an automated radio station and have contrasted the findings with an offline experiment. To our knowledge, this is the first study of this type in the context of session-based recommendation techniques.

Bearing in the mind that the number of participants might be a possible limitation of this work, the main findings of the study are as follows. First, using a comparably simple nearest-neighbor technique led to radio stations that were favorable, in more than one dimension, over a station that was built on a more complex deep learning technique. Second, the AR method, which recommended the most popular tracks, led to the highest number of likes, but was performing poorly in most other dimensions; optimizing and evaluating algorithms based on explicit like statements can therefore be misleading. Third, SPOTIFY's algorithm was better than all other methods in helping users discover new tracks they like. At the end, no differences in the participants' intention to re-use or recommend the system were found. The ability of SPOTIFY's algorithm to support discovery might therefore compensate other aspects where the algorithm did not excel. Fourth, the offline experiments indicate that precision and recall can be indicative of the quality of the recommendations to some extent. In general, however, the good quality perception of SPOTIFY's algorithm despite the very low offline accuracy results emphasizes that factors other than accuracy can be decisive for the long-term adoption of a system.

[5] Using means for *single* Likert-scale type items is considered potentially unreliable even though we only used labels for the ends of the numerical scale.
[6] A full table of the results is provided as auxiliary material in the ACM Digital Library.

REFERENCES

[1] Luke Barrington, Reid Oda, and Gert R. G. Lanckriet. 2009. Smarter than Genius? Human Evaluation of Music Recommender Systems. In *Proc. ISMIR '09*. 357–362.

[2] Jöran Beel and Stefan Langer. 2015. A Comparison of Offline Evaluations, Online Evaluations, and User Studies in the Context of Research-Paper Recommender Systems. In *Proc. TPDL '15*.

[3] Geoffray Bonnin and Dietmar Jannach. 2014. Automated Generation of Music Playlists: Survey and Experiments. *Comput. Surveys* 47, 2 (2014), 26:1–26:35.

[4] Shuo Chen, Josh L. Moore, Douglas Turnbull, and Thorsten Joachims. 2012. Playlist Prediction via Metric Embedding. In *Proc. KDD '12*. 714–722.

[5] Paolo Cremonesi, Franca Garzotto, and Roberto Turrin. 2012. Investigating the Persuasion Potential of Recommender Systems from a Quality Perspective: An Empirical Study. *Transactions on Interactive Intelligent Systems* 2, 2, Article 11 (June 2012), 11:1–11:41 pages.

[6] Michael D. Ekstrand, F. Maxwell Harper, Martijn C. Willemsen, and Joseph A. Konstan. 2014. User Perception of Differences in Recommender Algorithms. In *Proc. RecSys '14*. 161–168.

[7] Florent Garcin, Boi Faltings, Olivier Donatsch, Ayar Alazzawi, Christophe Bruttin, and Amr Huber. 2014. Offline and Online Evaluation of News Recommender Systems at Swissinfo.ch. In *Proc. RecSys '14*.

[8] Negar Hariri, Bamshad Mobasher, and Robin Burke. 2014. Context Adaptation in Interactive Recommender Systems. In *Proc. RecSys '14*. 41–48. http://doi.acm.org/10.1145/2645710.2645753

[9] Balázs Hidasi and Alexandros Karatzoglou. 2017. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. *CoRR* abs/1706.03847 (2017).

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

[11] Dietmar Jannach, Lukas Lerche, and Michael Jugovac. 2015. Item Familiarity as a Possible Confounding Factor in User-Centric Recommender Systems Evaluation. *i-com Journal of Interactive Media* 14, 1 (2015), 29–39.

[12] Iman Kamehkhosh, Geoffray Bonnin, and Dietmar Jannach. 2019. Effects of Recommendations on the Playlist Creation Behavior of Users. *User Modeling and User-Adapted Interaction* May (2019).

[13] Iman Kamehkhosh and Dietmar Jannach. 2017. User Perception of Next-Track Music Recommendations. In *Proc. UMAP '17*.

[14] Bart P. Knijnenburg, Martijn C. Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. 2012. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction* 22, 4 (2012), 441–504.

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

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

[17] Benedikt Loepp, Tim Donkers, Timm Kleemann, and Jürgen Ziegler. 2018. Impact of Item Consumption on Assessment of Recommendations in User Studies. In *Proc. RecSys '18*. 49–53.

[18] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of Session-based Recommendation Algorithms. *User-Modeling and User-Adapted Interaction* 28, 4–5 (2018), 331–390.

[19] Malte Ludewig, Iman Kamehkhosh, Nick Landia, and Dietmar Jannach. 2018. Effective Nearest-Neighbor Music Recommendations. In *Proc. RecSys '18 Challenge Workshop at ACM RecSys*.

[20] Andrii Maksai, Florent Garcin, and Boi Faltings. 2015. Predicting Online Performance of News Recommender Systems Through Richer Evaluation Metrics. In *Proc. RecSys '15*.

[21] Bamshad Mobasher, Honghua Dai, Tao Luo, and Miki Nakagawa. 2002. Using Sequential and Non-Sequential Patterns in Predictive Web Usage Mining Tasks. In *Proc. ICDM '02*. 669–672.

[22] Pearl Pu, Li Chen, and Rong Hu. 2011. A User-centric Evaluation Framework for Recommender Systems. In *Proc. RecSys '11*. 157–164.

[23] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. *Comput. Surveys* 51 (2018), 1–36. Issue 4.

[24] Marco Rossetti, Fabio Stella, and Markus Zanker. 2016. Contrasting Offline and Online Results when Evaluating Recommendation Algorithms. In *Proc. RecSys '16*.

[25] Guy Shani, David Heckerman, and Ronen I. Brafman. 2005. An MDP-Based Recommender System. *Journal of Machine Learning Researh* 6 (2005), 1265–1295.

[26] Longqi Yang, Michael Sobolev, Christina Tsangouri, and Deborah Estrin. 2018. Understanding User Interactions with Podcast Recommendations Delivered via Voice. In *Proc. RecSys '18*. 190–194.

[27] Yuan Yao and F. Maxwell Harper. 2018. Judging Similarity: A User-centric Study of Related Item Recommendations. In *Proc. RecSys '18*. 288–296.

[28] Yuan Yao and F. Maxwell Harper. 2018. Judging Similarity: A User-centric Study of Related Item Recommendations. In *Proc. RecSys '18*. 288–296.

[29] Qian Zhao, Martijn C. Willemsen, Gediminas Adomavicius, F. Maxwell Harper, and Joseph A. Konstan. 2018. Interpreting User Inaction in Recommender Systems. In *Proc. RecSys '18*. 40–48.