frontiers in Applied Mathematics and Statistics

MINI REVIEW
published: 29 August 2019
doi: 10.3389/fams.2019.00044

Deep Learning in Music Recommendation Systems

*Markus Schedl**

Institute of Computational Perception, Johannes Kepler University, Linz, Austria

Like in many other research areas, deep learning (DL) is increasingly adopted in music recommendation systems (MRS). Deep neural networks are used in this domain particularly for extracting *latent factors of music items* from audio signals or metadata and for learning *sequential patterns of music items* (tracks or artists) from music playlists or listening sessions. Latent item factors are commonly integrated into content-based filtering and hybrid MRS, whereas sequence models of music items are used for sequential music recommendation, e.g., automatic playlist continuation. This review article explains particularities of the music domain in RS research. It gives an overview of the state of the art that employs deep learning for music recommendation. The discussion is structured according to the dimensions of neural network type, input data, recommendation approach (content-based filtering, collaborative filtering, or both), and task (standard or sequential music recommendation). In addition, we discuss major challenges faced in MRS, in particular in the context of the current research on deep learning.

Keywords: music, recommender systems, music information retrieval, deep learning, neural networks, sequence-aware recommendation, automatic playlist continuation, survey

OPEN ACCESS

Edited by:
Michael Alexander Riegler,
Simula Research Laboratory, Norway

Reviewed by:
Junhui Wang,
City University of Hong Kong, Hong Kong
Li Su,
Academia Sinica, Taiwan

***Correspondence:**
Markus Schedl
markus.schedl@jku.at

Specialty section:
This article was submitted to Mathematics of Computation and Data Science, a section of the journal Frontiers in Applied Mathematics and Statistics

Received: *01 March 2019*
Accepted: *12 August 2019*
Published: *29 August 2019*

Citation:
Schedl M (2019) Deep Learning in Music Recommendation Systems. Front. Appl. Math. Stat. 5:44. doi: 10.3389/fams.2019.00044

1. INTRODUCTION

Research on music recommendation systems (MRS) is spiraling [1]. So is research in deep learning (DL). Despite their potential, neural network architectures are still surprisingly sparsely adopted for MRS, even though the number of respective publications is increasing. In this review article, we discuss the most recent research that involves DL in the context of MRS and we identify possible reasons for the still limited adoption of DL techniques in this recommendation domain.

Historically, research on MRS has emerged from two distinct communities, i.e., music information retrieval (MIR) and recommender systems (RS), with different focuses, perspectives, and terminologies.

1.1. Music Information Retrieval

MIR [2] has its origins in library science and signal processing [3]. It has therefore for a long time focused strongly on content-based approaches, where "content" refers to information extracted from the actual audio signal (compared with the common meaning of the term in RS research below). MIR research has created exciting tools and applications, e.g., musical score following [4, 5], intelligent music browsing interfaces [6, 7], or automatic music categorization (for instance, into genres [8–10] or affective categories such as mood [11, 12]), just to name a few. However, while much research in MIR has addressed the topic of audio similarity [13, 14], which is a prerequisite to build content-based MRS, surprisingly little research by the MIR community has been devoted specifically to music recommendation [15]. A simple quantitative investigation highlights this

fact: According to the computer science bibliography database DBLP[1] only 29 papers (or 1.6%) published at the main venue for MIR research, i.e., the ISMIR conference[2] between 2002 and 2018 (out of 1,840 total papers) contain the term "recommendation" or "recommender" in the title. For comparison, 120 papers (or 6.5%) contain the term "similarity" in their title.

1.2. Recommender Systems

RS research [16], in contrast, has traditionally been strongly driven by the task of movie recommendation, not least thanks to the Netflix Prize [17]. While collaborative filtering (CF) has been the most common choice in those early days of RS research, approaches based on content-based filtering (CBF) have gained popularity in recent years. In short, collaborative filtering approaches exploit interactions between users and items, e.g., clicks or ratings, which are represented in a user–item (rating) matrix R. The task is then to predict missing ratings $\hat{r}_{u,i}$ for pairs of users u and items i, and recommend to the target user u the (unseen) items with highest predictions. To this end, CF identifies similarities between users and/or items either in a low-dimensional joint representation of users and items (*model-based CF*) or by directly computing similarities from the user–item matrix (*memory-based CF*). In the latter case, we can distinguish between *user-based CF* and *item-based CF*, depending on whether recommendations are made based on similarities between users or between items. To give an example, user-based CF approaches often compute user similarities using Pearson's correlation coefficient (cf. Equation 1), where s_{uv} is the similarity between the item ratings given by users u and v; I_{uv} are the items both users u and v have rated; $\bar{r}_u$ ($\bar{r}_v$) is the mean rating of user u (v) and included to account for the user rating bias. A missing rating $\hat{r}_{ui}$ is then computed according to Equation (2), where N_u is the set of u's nearest neighbors (who rated item i) with respect to the similarity score s_u. Finally, items with highest $\hat{r}_{ui}$ are recommended to u.

$$s_{uv} = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt[2]{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2 \sum_{i \in I_{uv}} (r_{vi} - \bar{r}_u)^2}} \quad (1)$$

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{n \in N_u} s_{nu} \cdot (r_{ni} - \bar{r}_n)}{\sum_{n \in N_u} s_{nu}} \quad (2)$$

A common approach to content-based filtering is item-based nearest neighbors, where $\hat{r}_{ui}$ is determined by the ratings of u for similar items, for instance, as weighted average (cf. Equation 3). $N_u(i)$ in this case denotes the items most similar to item i which user u rated.

$$\hat{r}_{ui} = \frac{\sum_{n \in N_u(i)} s_{in} \cdot r_{ui}}{\sum_{n \in N_u(i)} s_{in}} \quad (3)$$

Note that this model for item-based CBF can also be adopted for item-based CF. In fact, the two differ only in their definition of s_{ij}. While item-based CF considers items i and j as similar if users have rated them in a similar way, item-based CBF considers them as similar if they share characteristics related to their content.

Comparing the MIR and RS communities with respect to their work on music recommendation, we note that already the use of the term "content" differs between these communities. While it indicates information extracted from the audio signal in MIR (such as rhythm, tempo, or melody), "content-based" RS have almost exclusively leveraged textual descriptors of the "content" (e.g., metadata, user-generated tags, or reviews) to effect recommendations [18]. This article also aims at raising awareness of such subtle differences between the MIR, RS, and other related communities such as information retrieval and multimedia. When it comes to music recommendation, the RS community, in particular represented by authors in the ACM Recommender Systems (RecSys)[3] conference proceedings, has embraced the emerging topic of MRS in the past few years. Quantitatively expressed, 43 papers (2.9%) published in all RecSys conferences (2007–2018) contain "music" in their title (out of 1,478 papers). The number of papers on MRS in ACM RecSys increased from only 2 in 2014 to 8 in 2018[4]. The enlarged interest for MRS is furthermore supported by the fact that the ACM RecSys Challenge 2018 targeted the topic of music playlist continuation [19].

Nowadays, DL in the domain of music recommendation is commonly used for (1) *automatic feature learning from audio signals* and creating corresponding embeddings for CBF, (2) modeling item/track sequences for *automatic music playlist continuation*, i.e., sequential music recommendation, and (3) *extracting latent factors from user–item rating data* to incorporate into CF models. This review article discusses approaches specific to the music domain. We therefore limit the scope to (1) and (2) and refrain from including general domain-independent work on the use of deep learning in purely CF-based approaches (3). For an up-to-date review of such research, we refer the reader to the survey by Batmaz et al. [20].

The article at hand is structured as follows: section 2 briefly summarizes in which ways music recommendation differs from recommendation in other domains and motivates the use of deep learning techniques to address some of these differences. Section 3 gives an overview about research that adopts DL for CBF or in hybrid systems that include a CBF component. Section 4 discusses work on sequence-aware music recommendation that employs DL. The article is round off by providing the author's personal opinion about the major current challenges in section 5. **Table 1** provides a compact overview of the research works discussed in sections 3 and 4. Since measures used in the evaluation of approaches strongly differ between discussed articles, we provide in **Table 2** a summary of the performance metrics adopted in the reviewed articles. Furthermore, **Table 3** lists abbreviations commonly used in DL and RS research, to offer the reader an easy reference.

Please note that this article does not provide explanations of the used neural network architectures and therefore addresses readers already familiar with these. The article instead offers a

[1] https://dblp.uni-trier.de
[2] https://www.ismir.net
[3] https://recsys.acm.org
[4] https://dblp.uni-trier.de/search?q=music%20venue%3ARecSys

TABLE 1 | Overview of discussed research articles.

Article	Year	Network type	Input data	CF	CBF	Seq.
Mcfee and Lanckriet [21]	2011	Markov chains	Timbre, tags, familiarity		✓	✓
Chen et al. [22]	2012	Markov embeddings	Playlists	✓		✓
van den Oord et al. [23]	2013	CNN	Spectrograms	✓	✓	
Wang and Wang [24]	2014	DBN	Spectrograms	✓	✓	
Liang et al. [25]	2015	MLP	MFCC	✓	✓	
Oramas et al. [26]	2017	MLP, CNN	Biographies, spectrograms		✓	
Sachdeva et al. [27]	2018	ANN, MLP, GRU	Sessions, tags		✓	✓
Vall et al. [28]	2018	RNN	Playlists	✓		✓
Vall et al. [29]	2019	CNN	Playlists, spectrograms, tags, listening logs		✓	✓
Lin et al. [30]	2019	ANN, RNN, GRU	Playlists, graphs, lyrics, album cover images		✓	✓

Column CF and CBF indicate whether the proposed solution uses collaborative filtering or content-based filtering (or both). Column Seq. identifies solutions that leverage sequential information in the training or prediction step.

TABLE 2 | Definitions and explanations of performance metrics reported in this article.

Metric	Definition	Explanation
Precision@K	$\frac{1}{\lvert U\rvert}\sum_{u\in U}\frac{\lvert L_u\cap\hat{L}_u\rvert}{\lvert\hat{L}_u\rvert}$	Percentage of top K recommended items which also appear as highly rated in the test set T, averaged over all users $u\in U$. Items in L_u are the relevant items (highly rated) for user u in T; $\hat{L}_u$ are the top K recommended items in T with highest predicted ratings for user u.
Recall@K	$\frac{1}{\lvert U\rvert}\sum_{u\in U}\frac{\lvert L_u\cap\hat{L}_u\rvert}{\lvert L_u\rvert}$	Percentage of items highly rated in the test set T that are recommended within the top K recommendations, averaged over all users $u\in U$. Terms are defined as in precision@K.
MAP@K	$\frac{1}{\lvert U\rvert}\sum_{u\in U}\left(\frac{1}{N}\sum_{i=1}^{K}P_u@i\cdot rel_u(i)\right)$	Mean average precision describes the overall precision at different numbers of recommended items, computed as arithmetic mean of the average precisions over all users U. Average precision of the top K recommendations of user u is computed using $P_u@i$, i.e. precision for user u at the i^{th} recommended item and $rel_u(i)$ which is an indicator signaling if the i^{th} recommended item is relevant ($rel(i)=1$) or not ($rel(i)=0$); N is the total number of relevant items. Note that MAP implicitly considers recall since the relevant items not in the recommendation list are included too.
AUC	$FPR=\frac{\lvert L_u^-\cap\hat{L}_u\rvert}{L_u^-}$, $TPR=recall$	Receiver operating characteristic (ROC) curve relates the false positive rate (FPR) to the true positive rate (recall), for all possible values of K recommendations. L_u^- are the irrelevant items (low-rated) for user u in T; $\hat{L}_u$ are the top K recommended items with highest predicted ratings for user u. The area under the ROC (AUC) can then be interpreted as the probability that the recommender will rank a randomly chosen positive (highly rated) item higher than a randomly chosen negative (disliked) one.
MRR	$\frac{1}{\lvert U\rvert}\sum_{u\in U}\frac{1}{rank_u}$	Mean reciprocal rank of user u is the inverse of the rank in u's recommendation list at which the first highly rated item occurs ($rank_u$). MRR is the average MRR_U computed over all users.
NDCG	$\frac{1}{\lvert U\rvert}\sum_{u\in U}\frac{\sum_{i=1}^{N}\frac{r_{u,i}}{log_2(i+1)}}{IDCG_u}$	Normalized discounted cumulative gain is a measure of the quality of the ranking of recommended items. Assuming that the items recommended for user u are sorted in decreasing order of their predicted rating, $r_{u,i}$ is the true rating of the item ranked at position i for user u; N is the length of the recommendation list; $IDCG_U$ represents the ideal DCG for user u, i.e., the DCG obtained if the recommended items were ordered exactly by their true ratings of u, in decreasing order.
RMSE	$\sqrt{\frac{1}{\lvert T\rvert}\sum_{r_{u,i}\in T}(r_{u,i}-\hat{r}_{u,i})^2}$	Root mean squared error between the predicted ratings $\hat{r}_{u,i}$ and the true ratings $r_{u,i}$ over all user–item pairs in the test set T.

compact overview of recent research to scholars and practitioners working on MIR and RS. For a more detailed introduction to DL, we refer to the books by Goodfellow et al. [31] or Aggarwal [32].

2. WHY MUSIC IS DIFFERENT

In comparison to other domains in which recommender systems are employed, such as products, movies, or hotels, recommendation in the music domain has certain specific characteristics that should be taken into account when creating MRS [1]. Some of these particularities in the music domain have implications on the use of recommender systems technology; and the use of deep learning approaches are directly motivated by them.

First, the *duration* of a music track is much shorter than the duration of a movie, holiday trip, or product usage. Second, the *number of items* in commercial music catalogs has a magnitude of tens of millions of tracks. For these two reasons, music might nowadays be considered *more disposable* than ever. Short consumption time and abundance of songs available implies that recommending a few songs that do not perfectly fit the user's taste

TABLE 3 | Abbreviations commonly used in this article and related literature.

Abbreviation	Full name
ANN	Attentive neural network
AUC	Area under the receiver operating characteristic curve
CNN	Convolutional neural network
CBF	Content-based filtering
CF	Collaborative filtering
CQT	Constant-Q transform
DNN	Deep neural network
DBN	Deep belief network
DL	Deep learning
GRU	Gated recurrent unit
k-NN	K-nearest neighbors
MAP	Mean average precision
MF	Matrix factorization
MFCC	Mel-frequency cepstral coefficient
MIR	Music information retrieval
MLE	Maximum likelihood estimation
MLP	Multi-layer perceptron
MRS	Music recommendation systems
MSD	Million Song Dataset
MSE	Mean squared error
NDCG	Normalized discounted cumulative gain
PCA	Principal components analysis
PMF	Probabilistic matrix factorization
ReLU	Rectified linear units
RMSE	Root mean squared error
RNN	Recurrent neural network
RS	Recommender systems
TF-IDF	Term frequency-inverse document frequency
VAE	Variational autoencoder
WMF	Weighted matrix factorization
WPE	Weighted prediction error

typically does not affect user experience in an overly negative way. This is in contrast to movie recommendation, for instance, where it takes users much longer to figure out that they dislike a recommended movie, and are therefore more upset about bad recommendations.

Third, *content-based features* extracted from the music audio signal historically play a much bigger role than in other domains, thanks to great advances in the research fields of music information retrieval and (audio) signal processing over the past decades. Building upon developed tools and gained knowledge from these fields, DL techniques can operate on a much larger and more sophisticated set of low- and mid-level audio features.

Fourth, *repeated* recommendations are sometimes appreciated by the listener, in contrast to the movie or product domain where users commonly disfavor recurring recommendations of the same items. Thanks to the probabilistic treatment of items in DL architectures, i.e., the network's output is usually a vector over items (or playlists) that contains the probabilities of fit, it is straightforward to include also already seen items.

Fifth, music has the power to send shivers down the listener's spine, i.e., music can evoke very strong *emotions*. State-of-the-art music emotion recognition techniques often make use of DL [11, 33]. Emotion-aware MRS then match the user's mood and the emotions evoked by songs in listeners, (e.g., Deng et al. [34]).

Sixth, music is often consumed in *sequence*, typically as playlists of music tracks or listening sessions. Therefore, recommending not only an unordered set of songs, but a meaningful sequence of songs, is an important task in the music domain. Since some DL techniques have particularly been developed to leverage sequential information, for instance, recurrent neural networks and their various extensions, their use greatly boosts approaches for automated playlist generation or next-track recommendation, cf. section 4.

Seventh, music consumption is often *passive*, i.e., the listener does not pay much attention to it, e.g., background music in shops or elevators. This can be critical when deriving positive implicit feedback, e.g., a song being played from beginning to end does not necessarily indicate that the listener actively consumed that song. Integrating additional contextual information, such as user's activity or apps interacted with while listening to music on smart devices, into context-aware MRS that nowadays are commonly powered by DL architectures is a solution to alleviate this problem.

3. CONTENT-BASED AND HYBRID APPROACHES

Recommender systems research in the music domain that leverages DL typically uses deep neural networks (DNN) to derive song or artist representations (embeddings or latent factors) from the audio content or textual metadata such as artist biographies or user-generated tags. These latent item factors are then either directly used in CBF systems such as nearest neighbor recommendation, integrated into matrix factorization approaches, or leveraged to build hybrid systems, most commonly integrating CBF and CF techniques.

Probably the earliest work that uses DL for content-based MRS is van den Oord et al.'s, who adopt a convolutional neural network (CNN) using rectified linear units (ReLU) and no dropout to represent each song by 50 latent factors learned from audio features [23]. As input, they use short music audio snippets retrieved from 7digital[5] for tracks in the Million Song Dataset (MSD) [35]. Training the CNN is then performed on log-compressed Mel spectrograms (128 frequency bands, window size of 23 ms, 50% window overlap), computed from randomly sampled 3-second-clips of the audio snippets. Two algorithmic variants are investigated: minimizing the mean squared error (MSE) and minimizing the weighted prediction error (WPE) as objective function. Experiments are conducted on 382 K songs and 1M users of the MSD. Play counts for user–song pairs are converted to binary implicit feedback data (i.e., 1 if user u listened to item i regardless of the listening frequency; 0 otherwise). Considering 500 predictions per user,

[5]https://www.7digital.com

the experiments show that a CNN using MSE as objective function performs best (AUC: 0.772). Linear regression using a common bag-of-audio-words representation (vector-quantized MFCCs) performed substantially worse (AUC: 0.645).

Wang and Wang use a deep belief network (DBN), mini-batch stochastic gradient descent and standard maximum likelihood estimation (MLE) for training [24]. Input data consists of randomly sampled 5-second-clips of audio snippets. Similar to van den Oord et al., Wang and Wang acquire these snippets from 7digital. The authors subsequently compute spectrograms (120 frequency bands) on windows of 30 ms (no overlap), resulting in a 166×120-matrix-representation of each 5-second-clip. Eventually, principal components analysis (PCA) is applied to reduce the dimensionality to 100 and this reduced signal representation is fed into the DBN. The item representations learned by the DBN are then integrated into a graphical linear model together with implicit user preferences. Evaluation is performed on the listening data of the top 100 K users in the MSD [35] who listened to 283 K unique songs. Play counts are converted into implicit feedback. The authors investigate a warm-start scenario (all users and all items in the test set also appear in the training set) and a cold-start scenario (all users but not all items appear in the training set). Using the root mean squared error (RMSE) as performance metric, the proposed DBN-based approach achieves 0.323 in warm-start and 0.478 in cold-start. Given the binary rating representation, a baseline that randomly predicts 0 or 1 would achieve an RMSE of 0.707; a mean predictor that always predicts a rating of 0.5 would achieve an RMSE of 0.500. Results of the DBN are almost equal to those achieved by the best-performing approach by van den Oord et al. [23], i.e., RMSE of 0.325 in warm-start and 0.495 in cold-start. Wang and Wang also propose a hybrid MRS that integrates the DBN output and a probabilistic matrix factorization model (PMF) [36] for collaborative filtering. This hybrid achieves an RMSE of 0.255 (warm-start).

Liang et al. propose a hybrid MRS that integrates content features learned via a multi-layer perceptron (MLP) as prior into probabilistic matrix factorization [25]. The authors train a MLP (3 fully-connected layers, ReLU activation, dropout, mini-batch stochastic gradient descent) using as input data vector-quantized MFCCs of 370K tracks of the MSD. The MLP is trained with the objective to predict 561 user-generated tags, i.e., for an auto-tagging or tag prediction task. The authors then use the output of the last hidden layer (1,200 units) as latent content representation of songs and assume that this representation captures music semantics. This latent content model is integrated as prior into a PMF model, which is trained with MLE. Evaluation is performed on subsets of the MSD for warm-start and cold-start (new items) situations on 614 K users and 97 K songs. In the warm-start scenario, Liang et al.'s hybrid approach using MLP and PMF performs equal to an approach that directly uses the vector-quantized MFCCs instead of training a MLP and also equal to a standard weighted matrix factorization (WMF) approach [37]; all achieve a normalized discounted cumulative gain (NDCG) of 0.288. The cold-start scenario illustrates that using the latent features given by the MLP clearly outperforms the sole use of MFCC features (NDCG of 0.161 vs. 0.143).

Oramas et al. propose an approach to create separate representations of music artists and of music tracks, and integrate both into a CBF system [26]. First, they use WMF on implicit feedback data (derived from play counts) to obtain latent factors for artists and for songs. Subsequently, DNNs are trained to learn the latent artist and the latent song factors independently, using as input artist and track embeddings created from artist biographies and song content, respectively. To create song embeddings, spectrograms are computed using the constant-Q transform (96 frequency bands, window size of 46 ms, no overlap). For each track, only one 15-second-snippet is considered. A CNN with ReLU activation and 50% dropout trained on the fixed-length CQT patches is then used to compute track embeddings. Artist embeddings are learned from biographies enriched with information from the DBpedia[6] knowledge graph and represented as term frequency-inverse document frequency (TF-IDF) feature vectors [38]. A MLP using as input these TF-IDF vectors is then trained to obtain latent artist factors. The artist and track features are finally combined in a late fusion fashion, using again a MLP. Evaluation is carried out on a subset of the MSD (329 K tracks by 24 K artists for which biographies and audio are available). Oramas et al. report mean average precision (MAP) values at 500 recommendations of up to 0.020 for their approach when evaluated in an artist recommendation task and up to 0.004 when recommending tracks.

4. SEQUENCE-AWARE MUSIC RECOMMENDATION

Listening to music is an inherently sequential process. Music aficionados and professional music editors create carefully hand-crafted playlists for specific purposes or featuring a common theme, cf. [39, 40]. Such playlists as well as users' *ad-hoc* listening sessions can be leveraged to build sequence models of tracks or artists that are in turn used for MRS. Thanks to recent efforts such as the ACM Recommender Systems Challenge 2018[7] [19, 41] or the Sequential Skip Prediction Challenge of the WSDM Cup 2019,[8] research on sequence-aware MRS is experiencing a boost. Approaches taken in the ACM Recommender Systems Challenge 2018 are summarized and discussed in Lee [41].

Most research on the topic targets either *next track recommendation* (also known as next song recommendation) or, more generally, *automatic playlist continuation* (APC), cf. [42, 43]. Both tasks constitute of learning a model from existing playlists or listening sessions, but they differ in terms of output. While APC approaches produce track sequences of arbitrary length, next track recommenders only suggest one track to listen to next.

For a systematic investigation of early works on APC, we recommend the survey and experiments conducted by Bonnin and Jannach [42]. They assess on common datasets a set of non-DL-based methods, i.e., *k*-nearest neighbors (*k*-NN), association

[6] https://www.dbpedia.org
[7] http://www.recsyschallenge.com/2018
[8] http://www.wsdm-conference.org/2019/wsdm-cup-2019.php

rules, sequential patterns, and three other simple approaches: adding to the playlist the most popular artists in the training set in decreasing order of overall playlist occurrence, adding only tracks by the same artists that already appear in the given playlist and order them by popularity ("Same Artist—Greatest Hits"), and adding tracks by the same artists already included in the playlist and by similar artists where similarity is defined via artist co-occurrence in other playlists ("Collocated Artists — Greatest Hits"). The authors compare these approaches in terms of recall at different numbers n of recommended items. They find that simple k-NN outperforms the other approaches for small values of n, but the Collocated Artists—Greatest Hits approach outperforms all others for larger n values, depending on the dataset starting as early as $n \approx 50$ (50 K playlists retrieved from the streaming service Last.fm) up to as late as $n \approx 1,700$ (for a subset of 50 K playlists of the AotM-2011 dataset [21]).

Earliest works that leverage song orderings within playlists predominantly use Markov chain models [21, 22]. While not strictly adopting a neural network model, we mention these works here for historical reasons because they mark the beginning of research on APC. McFee and Lanckriet use a generative model trained on hand-curated playlists, i.e., the AotM-2011 dataset [21]. The authors represent songs by audio content features (timbre descriptors), tags, and estimates of familiarity, adopting ideas from statistical natural language processing. They train various Markov chains to model transitions between songs and use them for generating new songs that fit a given playlist. Another early approach, which does not rely on audio features, is proposed by Chen et al. [22]. The authors use logistic Markov embeddings to model song transitions, which are learned from hand-curated training playlists. This approach resembles matrix factorization and results in an embedding of songs in Euclidean space based on which nearest songs to the songs in a given playlist can be easily identified. In these early works [21, 22], performance is measured in terms of log-likelihood of the model to produce the actual, known playlist continuations.

More recent work includes Vall et al.'s who target the next track scenario and compare different approaches with respect to their ranking performance [28]. The authors analyze the influence of length of the user-generated playlists used for training and of the order of songs in the training playlists on prediction performance. They use a recurrent neural network (RNN), adopted from [44], that only takes sequential information without any additional metadata into account. Evaluation is conducted based on two datasets: AotM-2011 [45] and a commercial dataset provided by 8tracks[9], a streaming service for user-generated playlists. The authors use subsets of AotM-2011 and 8tracks, the former containing 2.7 K playlists with 12.3 K songs, the latter 3.3 K playlists with 14.6 K songs. To measure performance, they compute the rank in the list of predictions at which the actual next song according to the ground truth occurs. Compared to a popularity-based recommender and an item-based CF system, the RNN yields more stable results and is unaffected by popularity bias. Its performance further converges already when given as input playlists of short length (2 or 3 songs, depending on the dataset used). The CF system performs clearly worst. The popularity-based system achieves remarkably good results similar to those of the RNN. Song order in training and test playlists is not found to significantly influence the performance of the RNN.

In a follow-up work, Vall et al. propose two approaches to APC: profile-based and membership-based playlist continuation [29]. The former classifies each song according to whether it fits to each playlist in the catalog and subsequently ranks, for a given playlist p, the candidate songs with respect to their predicted probability of fit to p. It takes arbitrary feature vector representations of songs as input to train a DNN, minimizing binary cross-entropy loss. The membership-based approach represents not only songs but also playlists as feature vectors (of their constituting songs). In addition to the song vectors, like in the profile-based approach, also the playlist features are transformed into latent factors, i.e., a matrix of latent factors, one row for each song in the playlist. This is achieved via a DNN whose output is averaged over the latent factor representations of all songs in the playlist. Both songs and playlists are therefore represented in the same vector space, which allows to directly apply a distance metric to estimate their goodness of match. Again, binary cross-entropy loss is used as objective function when training the DNN. Profile-based APC can deal with the new item (song) problem, i.e., songs not seen during training can be recommended. Membership-based APC can additionally be used for new playlists, i.e., playlists unseen at training time can be extended. As DNN, the authors use a CNN to obtain 200-dimensional song (or playlist) representations using an approach similar to van den Oord et al.'s [23]. They also assess text embeddings of Last.fm tags created by word2vec [46] as well as WMF [37] applied to listening logs from the MSD. Evaluation is carried out like in 40. McFee and Lanckriet [28] on the AotM-2011 and 8tracks subsets. When representing songs by a concatenation of the CNN, word2vec, and WMF features, the profile-based approach accomplishes a recall@100 of 0.178; the membership-based approach reaches 0.181.

Sachdeva et al. propose an attentive neural network (ANN) architecture for next song prediction [27]. They use one-hot encoded representations of songs and of tags preceding the song to be predicted. Tags and listening sessions are crawled from Last.fm, including 386 K songs (no more details are given). In the proposed ANN architecture, sequential song and tag representations are fed into bidirectional gated recurrent units (GRU), whose result is combined in the attention layer. The attention layer's context vector for songs and tags is then concatenated and fed into a MLP, using a softmax function to create a probability distribution over songs to serve as the next song to play. Using only song representations, Sachdeva et al.'s approach achieves a recall@10 of 0.264. Using song and tag representations, recall@10 increases to 0.299.

Lin et al. propose another variant of an ANN which they name heterogeneous knowledge-based attentive neural networks (HK-ANN) [30]. This architecture contains two components: entity embedding and short-term recommendation. The embedding part learns embeddings from 3 different multimodal sources,

[9]https://8tracks.com

i.e., graph data, textual data, and visual data. Graph data is represented by a graph connecting songs, albums, artists, users, playlists, and tags; textual data by song lyrics; visual data by cover images. Graphical, textual, and visual embeddings are learned separately. Graph embeddings are created using TansR [47] for heterogeneous networks, textual embeddings using paragraph vector with distributed memory (PV-DM) [48], and visual embeddings by a variational autoencoder (VAE) [49]. Each of the 3 embeddings are created with a dimensionality of 100. They are concatenated to form a 300-dimensional vector representation. These embeddings constitute the input to the short-term recommendation component of the architecture. More precisely, users' short-term listening sequences of songs are represented by the corresponding embeddings created in the entity embedding step described above and fed into a RNN with bidirectional GRU. In the encoding stage, the output of the RNN is input into an attention layer from which the decoding stage (again RNN with bi-GRU) linearly combines the parts of the input sequence. This linear combination represents an attention mechanism whose output is enriched by the latent factors of candidate song items and of the current user, both represented by the embedding created in the first stage of the algorithm. The decoding stage therefore integrates the users' attention of sequences, their embeddings, and the embeddings of candidate song items. Prediction is eventually effected using a softmax layer. For training, mini-batch stochastic gradient descent is used to minimize cross-entropy loss. The authors evaluate their approach on a sample drawn from NetEase Cloud Music[10] a Chinese music streaming platform. The dataset comprises 9 K users, 1.4 M songs, 421 K albums, 151 K artists, and 100 K playlists as well as a large number of relational data to create the network used in the graph embedding step. For more details, please refer to Lin et al. [30]. The authors report several performance measures at 20 recommended items: recall of 0.425, MRR of 0.305, and NDCG of 0.385. The proposed HK-ANN model thereby outperforms several traditional and state-of-the-art approaches used as baseline. However, these baselines do not take advantage of the full range of multimedia material used in the HK-ANN approach.

5. CURRENT CHALLENGES

Despite its recent increasing adoption, current research on deep learning for MRS is facing several challenges. First, as with many tasks that employ DL, *transparency* is an issue. In the domain of music recommendation, this relates to the transparency of why a MRS decides whether a particular music item is suggested to the target user, i.e., to the system's reasoning when making predictions. While several traditional machine learning methods, such as rule learners or decision trees, are capable of providing explanations for their classification or regression decisions, the latent factors or embeddings used in current DL-based MRS do not offer meaningful explanations. One direction to alleviate this problem is to integrate traditional supervised methods with DNN, as proposed for instance in Zhang et al. [50] and Laptev and Buhmann [51] as combinations of CNN and decision trees.

Another major challenge resulting from the recent fast growing application of DL in MRS is the *lack of established multimodal datasets* and the large *variety of evaluation metrics* used to compare results between approaches. While some real-world datasets such as the MSD [35] or the LFM-1b [52] dataset do exist, several authors (have to) stick to other datasets, usually self-assembled, because their approaches require additional data not included in existing datasets, e.g., full lyrics, audio content, or images of album covers. Furthermore, the used evaluation metrics and their parameters (e.g., number of recommended items) strongly vary between research works, which can be well observed from the review provided in the previous sections. Therefore, there is an urgent need for multimodal datasets for MRS as well as for establishing a common agreement on evaluation metrics to be used for common tasks in MRS, for instance, APC.

Moreover, a shortcoming that particularly affects sequence-aware MRS is that most research takes a highly system-centric view. In particular in DL-based approaches, large amounts of playlist data are leveraged to create computational sequence models, without understanding, or considering in the model the user's intention to create a playlist or the purpose of a listening session. Similarly, little is known about the relevance of playlist properties, such as song order or diversity, for users. Only recently, researchers started to investigate these questions, cf. [53, 54]. Notwithstanding the importance of a system-centric perspective, the challenge is to achieve a *balance between a purely system-centric view and a holistic user-centric view*. The latter should be considered both when devising algorithms (e.g., by integrating holistic user models) and when evaluating MRS powered by these algorithms (e.g., by adopting beyond-accuracy metrics and assessing the user experience).

Eventually, several other challenges relate more closely to the particularities of the music domain. One is to answer the question of how to deal with *different variants of the same music piece* (e.g., cover songs or different interpretations of a piano sonata). Depending on the target user's preferences for or against certain performers, the answer to this question may have a strong influence on the perceived quality of recommendations. Also, *complex recommendation scenarios* emerged recently in the music domain. One remarkable example is that of sheet music recommendation for piano practitioners as offered, for instance, by OKTAV[11]. In such a scenario, the recommendation engine needs to consider not only the general music preferences of the user, but more specifically also his or her playing preferences and, most importantly, piano skills to create personalized recommendations. Another scenario is group recommendation, for instance, how to create a playlist to listen to in a car with several passengers. This scenario not only requires to take into account the situational aspects, such as weather, road condition, or traffic intensity, but also to consider multiple passengers' preferences and align them with the driver's. A final challenge is

[10] https://www.music.163.com

[11] https://www.oktav.com

that of *multi-stakeholder music recommendation*. Current MRS are often tuned to benefit a single stakeholder (e.g., shareholders of the streaming company, content creators, content providers, or music consumers), whereas multi-stakeholder recommenders take a holistic perspective onto the needs and demands of the individual stakeholders and consider them jointly (e.g., increasing the business value of a company, raising the popularity of a creator's work, or discovering interesting new music for the consumer).

AUTHOR CONTRIBUTIONS

The author confirms being the sole contributor of this work and has approved it for publication.

FUNDING

This work was supported by the Austrian Science Fund (FWF): P25655.

REFERENCES

1. Schedl M, Zamani H, Chen CW, Deldjoo Y, Elahi M. Current challenges and visions in music recommender systems research. *Int J Multi Inform Ret*. (2018) **7**:95–116. doi: 10.1007/s13735-018-0154-2
2. Schedl M, Gómez E, Urbano J. Music information retrieval: recent developments and applications. *Found Trends Inform Ret*. (2014) **8**:127–261. doi: 10.1561/1500000042
3. Downie JS. Music information retrieval. *Ann Rev Inform Sci Techn*. (2003) **37**:295–340. doi: 10.1002/aris.1440370108
4. Dorfer M, Henkel F, Widmer G. Learning to listen, read, and follow: score following as a reinforcement learning game. In: *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR 2018)*, Paris (2018). p. 784–91.
5. Chou PW, Lin FN, Chang KN, Chen HY. A simple score following system for music ensembles using chroma and dynamic time warping. In: *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval (ICMR 2018)*, Yokohama: ACM (2018). p. 529–32. doi: 10.1145/3206025.3206090
6. Goto M, Dannenberg RB. Music interfaces based on automatic music signal analysis: new ways to create and listen to music. *IEEE Signal Proc Mag*. (2019) **36**:74–81. doi: 10.1109/MSP.2018.2874360
7. Schedl M. Intelligent user interfaces for social music discovery and exploration of large-scale music repositories. In: *Proceedings of the 22nd ACM International Conference on Intelligent User Interfaces (IUI 2017): Workshop on Theory-Informed User Modeling for Tailoring and Personalizing Interfaces (HUMANIZE 2017)*. Limassol: ACM (2017). p. 7–11. doi: 10.1145/3039677.3039678
8. Oramas S, Nieto O, Barbieri F, Serra X. Multi-label music genre classification from audio, text and images using deep features. In: *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou (2017). p. 23–30.
9. Mayer R, Rauber A. Music genre classification by ensembles of audio and lyrics features. In: *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*. Miami, FL (2011). p. 675–80.
10. Sturm BL. Classification accuracy is not enough: on the evaluation of music genre recognition systems. *J Intell Inform Syst*. (2013) **41**:371–406. doi: 10.1007/s10844-013-0250-y
11. Yang YH, Chen HH. Machine recognition of music emotion: a review. *Trans Intell Syst Techn*. (2013) **3**:40. doi: 10.1145/2168752.2168754
12. Huq A, Bello JP, Rowe R. Automated music emotion recognition: a systematic evaluation. *J New Music Res*. (2010–11) **39**:227–44. doi: 10.1080/09298215.2010.513733
13. Knees P, Schedl M. Music similarity and retrieval — an introduction to audio- and web-based strategies. Berlin; Heidelberg: Springer (2016).
14. Karydis I, Lida Kermanidis K, Sioutas S, Iliadis L. Comparing content and context based similarity for musical data. *Neurocomputing*. (2013) **107**:69–76. doi: 10.1016/j.neucom.2012.05.033
15. Schedl M, Knees P, McFee B, Bogdanov D, Kaminskas M. Music recommender systems. In: Ricci F, Rokach L, Shapira B, Kantor PB, editors. *Recommender Systems Handbook, 2nd ed*. Boston, MA: Springer (2015). p. 453–92.
16. Ricci F, Rokach L, Shapira B, Kantor PB, editors. *Recommender Systems Handbook, 2nd ed*. Boston, MA: Springer (2015).
17. Bell RM, Koren Y. Lessons from the Netflix Prize challenge. *ACM SIGKDD Exp Newslett*. (2007) **9**:75–9. doi: 10.1145/1345448.1345465
18. de Gemmis M, Lops P, Musto C, Narducci F, Semeraro G. In: Ricci F, Rokach L, Shapira B, Kantor PB, editors. *Semantics-Aware Content-Based Recommender Systems, 2nd ed*. Boston, MA: Springer (2015). p. 119–59.
19. Chen CW, Lamere P, Schedl M, Zamani H. RecSys Challenge 2018: Automatic Music Playlist Continuation. In: *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys 2018)*. Vancouver, BC: ACM (2018). p. 527–8.
20. Batmaz Z, Yurekli A, Bilge A, Kaleli C. A review on deep learning for recommender systems: challenges and remedies. *Artif Intell Rev*. (2018) **52**:1–37. doi: 10.1007/s10462-018-9654-y
21. McFee B, Lanckriet G. The natural language of playlists. In: *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*. Miami, FL (2011).
22. Chen S, Moore JL, Turnbull D, Joachims T. Playlist prediction via metric embedding. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. Beijing (2012).
23. van den Oord A, Dieleman S, Schrauwen B. Deep content-based music recommendation. In: Burges C, Bottou L, Welling M, Ghahramani Z, Weinberger K, editors. *Advances in Neural Information Processing Systems 26 (NIPS)*. Lake Tahoe, NV: Curran Associates, Inc. (2013). p. 2643–51.
24. Wang X, Wang Y. Improving content-based and hybrid music recommendation using deep learning. In: *Proceedings of the 22nd ACM International Conference on Multimedia*. Orlando, FL: ACM (2014). p. 627–36. doi: 10.1145/2647868.2654940
25. Liang D, Zhan M, Ellis DPW. Content-aware collaborative music recommendation using pre-trained neural networks. In: *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR 2015)*. (2015). p. 295–301.
26. Oramas S, Nieto O, Sordo M, Serra X. A deep multimodal approach for cold-start music recommendation. In: *Proceedings of the 2Nd Workshop on Deep Learning for Recommender Systems. DLRS 2017*. New York, NY: ACM (2017). p. 32–7. doi: 10.1145/3125486.3125492
27. Sachdeva N, Gupta K, Pudi V. Attentive neural architecture incorporating song features for music recommendation. In: *Proceedings of the 12th ACM Conference on Recommender Systems. RecSys '18*. New York, NY: ACM (2018). p. 417–21. doi: 10.1145/3240323.3240397
28. Vall A, Quadrana M, Schedl M, Widmer G. The importance of song context and song order in automated music playlist generation. In: *Proceedings of the 15th International Conference on Music Perception and Cognition (ICMPC) and 10th Triennial Conference of the European Society for the Cognitive Sciences of Music (ESCOM)*. Graz (2018).
29. Vall A, Dorfer M, Eghbal-Zadeh H, Schedl M, Burjorjee K, Widmer G. Feature-combination hybrid recommender systems for automated music playlist continuation. *User Model User Adapted Interact*. (2019) **29**:527–72. doi: 10.1007/s11257-018-9215-8.
30. Lin Q, Niu Y, Zhu Y, Lu H, Mushonga KZ, Niu Z. Heterogeneous knowledge-based attentive neural networks for short-term music recommendations. *IEEE Access*. (2018) **6**:58990–9000. doi: 10.1109/ACCESS.2018.2874959
31. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. Cambridge, MA: MIT Press (2016).

32. Aggarwal CC. *Neural Networks and Deep Learning: A Textbook*. Springer International Publishing (2018).
33. Yang X, Dong Y, Li J. Review of data features-based music emotion recognition methods. *Multimedia Syst*. (2018) **24**:365–89. doi: 10.1007/s00530-017-0559-4.
34. Deng S, Wang D, Li X, Xu G. Exploring user emotion in microblogs for music recommendation. *Expert Syst Appl*. (2015) **42**:9284–93. doi: 10.1016/j.eswa.2015.08.029
35. Bertin-Mahieux T, Ellis DPW, Whitman B, Lamere P. The million song dataset. In: *Proceedings of the 12th International Society for Music Information Retrieval Conference*. Miami, FL (2011). p. 591–6.
36. Salakhutdinov R, Mnih A. Probabilistic matrix factorization. In: *Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS 2007)*. Vancouver, BC: Curran Associates Inc. (2007). p. 1257–64. Available online at: http://dl.acm.org/citation.cfm?id=2981562.2981720
37. Hu Y, Koren Y, Volinsky C. Collaborative filtering for implicit feedback datasets. In: *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM)*. Washington, DC (2008). doi: 10.1109/ICDM.2008.22
38. Baeza-Yates R, Ribeiro-Neto B. *Modern Information Retrieval – The Concepts and Technology Behind Search, 2nd ed*. Harlow: Addison-Wesley, Pearson (2011).
39. Lee JH. How similar is too similar? exploring users' perceptions of similarity in playlist evaluation. In: *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*. Miami, FL (2011).
40. Cunningham SJ, Bainbridge D, Falconer A. "More of an art than a science": supporting the creation of playlists and mixes. In: *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*. Victoria, BC (2006).
41. Zamani H, Schedl M, Lamere P, Chen CW. An analysis of approaches taken in the ACM RecSys challenge 2018 for automatic music Playlist continuation. *arXiv*. (2018) *arXiv:1810.01520*.
42. Bonnin G, Jannach D. Automated generation of music playlists: survey and experiments. *ACM Comput Surv*. (2014) **47**:26. doi: 10.1145/2652481
43. Quadrana M, Cremonesi P, Jannach D. Sequence-aware recommender systems. *ACM Comput Surv*. (2018) **51**:1–66. doi: 10.1145/3190616
44. Hidasi B, Karatzoglou A, Baltrunas L, Tikk D. Session-based recommendations with recurrent neural networks. *CoRR*. (2015) abs/1511.06939.
45. McFee B, Lanckriet G. Hypergraph models of playlist dialects. In: *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*. Porto (2012).
46. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. NIPS'13*. Lake Tahoe, NV Curran Associates Inc. (2013). p. 3111–9. Available online at: http://dl.acm.org/citation.cfm?id=2999792.2999959
47. Lin Y, Liu Z, Sun M, Liu Y, Zhu X. Learning entity and relation embeddings for knowledge graph completion. In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. AAAI'15*. Austin, TX: AAAI Press (2015). p. 2181–7. Available online at: http://dl.acm.org/citation.cfm?id=2886521.2886624
48. Le Q, Mikolov T. Distributed representations of sentences and documents. In: *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32. ICML'14*. JMLR.org (2014). p. II–1188–96. Available online at: http://dl.acm.org/citation.cfm?id=3044805.3045025
49. Kingma DP, Welling M. Auto-encoding variational bayes. *CoRR*. (2013) abs/1312.6114.
50. Zhang Q, Yang Y, Wu YN, Zhu S. Interpreting CNNs via decision trees. *CoRR*. (2018) abs/1802.00121.
51. Laptev D, Buhmann JM. Convolutional decision trees for feature learning and segmentation. In: Jiang X, Hornegger J, Koch R, editors. *Pattern Recognition*. Cham: Springer (2014). p. 95–106.
52. Schedl M. The LFM-1b dataset for music retrieval and recommendation. In: *Proceedings of the 6th ACM International Conference on Multimedia Retrieval (ICMR 2016)*. New York, NY: ACM (2016). p. 103–10. doi: 10.1145/2911996.2912004
53. Kamehkhosh I, Jannach D, Bonnin G. How automated recommendations affect the playlist creation behavior of users. In: *Joint Proceedings of the 23rd ACM Conference on Intelligent User Interfaces (ACM IUI 2018) Workshops: Intelligent Music Interfaces for Listening and Creation (MILC)*. Tokyo (2018).
54. Tintarev N, Lofi C, Liem CCS. Sequences of diverse song recommendations: an exploratory study in a commercial system. In: *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization. UMAP '17*. New York, NY: ACM (2017). p. 391–2. doi: 10.1145/3079628.3079633

Conflict of Interest Statement: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.