PLAYLISTS FROM THE MATRIX — COMBINING AUDIO AND METADATA IN SEMANTIC EMBEDDINGS

Kyrill Pugatschewski[1,2] **Thomas Köllmer**[1] **Anna M. Kruspe**[1]

[1] Fraunhofer IDMT, Ilmenau, Germany
[2] University of Saarland, Saarbrücken, Germany
s9kypuga@stud.uni-saarland.de, kor@idmt.fhg.de, kpe@idmt.fhg.de

ABSTRACT

We present a hybrid approach to playlist generation which combines audio features and metadata. Three different matrix factorization models have been implemented to learn embeddings for audio, tags and playlists as well as factors shared between tags and playlists. For training data, we crawled Spotify for playlists created by tastemakers, assuming that popular users create collections with high track cohesion. Playlists generated using the three different models are presented and discussed.

1. INTRODUCTION

The overabundance of digital music makes manual compilation of good playlists a difficult process. How to find the best possible tracks which all fit one mood, like *songs for rainy evenings*?

Playlist generation is nowadays a common feature of streaming services, Spotify's *Discover Weekly* [1] being a prominent example. But though playlists may be based on musical and acoustic similarities between tracks as well as themes such as *90s* or *European Hits*, most approaches to playlist generation focus either on audio [2] or on metadata [3]. Applying ideas from hybrid music recommendation, we propose a framework based on matrix factorization which combines audio features and metadata to predict playlists for tracks. Starting from user-selected seed tracks, we use that model to bootstrap playlists through self-learning.

[1] http://www.spotify.com/is/discoverweekly

2. DATASET

2.1 Crawling

The Spotify Web API [2] has been chosen as source of playlist information. We focused on playlists created by tastemakers, popular users such as labels and radio stations, assuming that their track listings have higher cohesiveness in comparison to user collections of their favorite songs. Extracted data includes 30-second preview clips, genres and high-level audio features such as danceability and instrumentalness.

In total, we crawled 11,031 playlists corresponding to 4,302,062 distinct tracks. For 1,147 playlists, we have preview clips for their 54,745 distinct tracks. 165 playlists are curated by tastemakers, corresponding to 6,605 tracks.

2.2 Preprocessing

To use the high-level audio features within our model, we discretized them into tags by simple rules. E.g. a value of speechiness between 0.33 and 0.66 becomes the *rap*, valence and arousal values are mapped to mood descriptors according to the circumplex model of affect. [4]

The 1322 Spotify genres we obtained through crawling are mapped to a list of 13 predefined genres. This is done by linking them to DBpedia [1] and searching its underlying ontology for supergenres which match one of the given genres.

The track's audio information is represented by codebook vectors based on MFCCs extracted from the preview clips.

3. RECOMMENDER MODEL

The general framework for learning a playlist recommendation model is inspired by Weston et al. [5]:
A low-dimensional embedding optimizing precision at k of the ranked list of retrieved entities is learned using the stochastically optimized Weighted Approximately Ranked Pairwise (WARP) loss. Playlists ranks for a track are determined by the product of the track's feature vector and the factorized model. Within this framework, three models have been implemented.

[2] http://developer.spotify.com/web-api

Features Approach

A simple ranking function is learned as a factorization into playlist embeddings and an audio embedding matrix. This corresponds to a direct mapping between playlists and features.

Top-*k* Tags Appproach

A tag ranking function as factorization into tag embeddings and audio embedding is combined with a playlist ranking as factorization into playlist embeddings and tag embeddings. First, for playlist prediction, the k best tags are determined for a track. Playlist scores are then calculated and summed up for each of the k tags. This gives the track's playlists scores. Here, tags are treated as high-level semantic summaries of a track.

Tri-factorization approach

A tag ranking function as factorization into tag embeddings, shared latent factors and audio embedding is learned as well as a playlist ranking function as factorization into playlist embeddings, shared latent factors and audio embedding. Both rankings are learned together but only the playlist ranking is used for evaluation. The shared latent factors are assumed to capture common reasoning which underlies the description of a track by tags and membership in a playlist.

4. PLAYLIST GENERATION

On the input side, the generator takes positive and negative seed tracks around which the playlist will be built. Positive seeds steer the overall direction of the playlist, negative seed tracks give examples of undesired content. Audio, tag and shared playlist-tag embeddings are taken from a previously trained model. The positive and negative playlist embeddings are determined through self-learning using one of the three aforementioned approaches starting from the seeds.

5. DISCUSSION & FUTURE WORK

Formal evaluation of playlists generated from the three models is yet to be done. A user study in which participants can rank different playlists may inform a meaningful metric for track cohesion.

The top-k tags approach seems to perform worst. Generated playlists feature up to several tracks similar to the negative seeds (see Table 1,2: Gojira and Sister Sin both play Metal, Earth Crisis play Hardcore Punk) which may be due to the simplistic method we chose for conversion of high-level audio features to tags. A proper model for tag mapping could improve the annotation quality.

Though the features approach includes less negative tracks, the playlists feel too diverse (e.g. Kraftwerk and Bill Withers). This may be caused by the high musical diversity in pop music which the MFCC based features cannot fully capture. Tri-factorization gives probably the most balanced playlist, Cartoon is Electro but the rest of the tracks feels less far apart than with the features approach.

Features	The Jacksons - State of Shock
	Kraftwerk - The Model
	Earth Crisis - New Ethic
	Bill Withers - Lovely Day
	The Forecast - Clear Eyes
Top-k tags	C. Aguilera - Shut Up
	Earth Crisis - Smash or Be Smashed
	B. Springsteen - Incident on 57th Street
	Distrion - Entropy
	Sister Sin - I'm Not You
Tri-factorization	The Isley Brothers - Caravan of Love
	Enrique Iglesias - Hero
	Mahmundi - Azul
	Cartoon - On & On
	Straylight Run - Existentialism

Table 1. Example playlists generated by the three approaches outlined in section 3. The used seed songs are listed in table 2.

Positive seeds	Negative seeds
Rihanna - Work	Bedrich Smetana - Má vlast
Bruno Mars - Treasure	Gojira - L'Enfant Sauvage
Ariana Grande - Be Alright	Jack Ü - Take Ü There

Table 2. Example of positive and negative seeds for playlist generation.

6. REFERENCES

[1] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. Dbpedia a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195, 2015.

[2] F. Maillet, D. Eck, G. Desjardins, and P. Lamere. Steerable playlist generation by learning song similarity from radio station playlists. In *Proceedings of the International Symposium on Music Information Retrieval*, pages 345–350, 2009.

[3] J. C. Platt, C. J. C. Burges, S. Swenson, C. Weare, and A. Zheng. Learning a gaussian process prior for automatically generating music playlists. In *Advances in Neural Information Processing Systems*, pages 1425–1432, 2001.

[4] J. A. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178, 1980.

[5] J. Weston, S. Bengio, and P. Hamel. Multi-tasking with joint semantic spaces for large-scale music annotation and retrieval. *Journal of New Music Research*, 40(4):337–348, 2011.