The Geometry of Playlists: Navigating the Spaces of Songs to Build Playlists

Team Spotify: Mehul Smriti Raje, Benjamin Sanchez-Lengeling, Timothy Lee, Spandan Madan
AC 297r: Computational Science and Engineering Capstone, Spring 2018

1. Intro and Task

Spotify offers curated playlists to help variety of users with different interests, moods, demographics find the right music. Automatic playlist generation would aid this purpose.

Task:
Generating tracks to continue an incomplete playlist, assuming it has the same intent throughout.

2. Data

- Comprehensive 1 million playlists dataset (Recsys challenge) with each playlist having 66 tracks on average
- 2.2 million unique tracks, 300k unique artists
- Largely US based, young people
- To understand automatic playlist continuation, we add data about lyrics, annotations from Genius.com and data collected from UX experiment

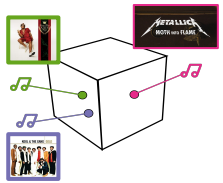
Inside a track:

Artist: Bruno Mars
Album: 24K Magic
Genres: Contemporary R&B, New Jack Swing, Hip Hop Soul
Popularity: 100

Lyrics
Audio Features: *Danceability, Energy, Speechiness, Tempo , etc.*
Genius annotations: **"***The song starts out acapella with a vocoder/talk box effect…"*

3. Model

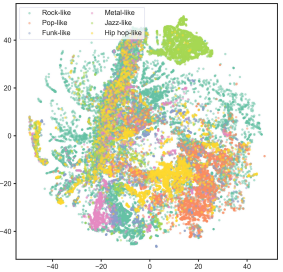
Embedding



Music has many dimensions. We group it based on emotions it evokes, style and genre, artists and popularity.

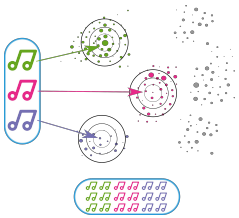
Vector representation/ embeddings are built for each data source. This creates vectors in a continuous space with natural distances, which are used to search for similar objects.

t-SNE in track space, colored by genre-topic



Pooling

Reduces our search space from 2.2 million tracks to 10k tracks. We select tracks based on similarity to the initial set.



Ranking

Using distances in each type of feature we learn the ideal mix of information needed to reconstruct the normal ranking of a playlist using bayesian optimization,

$$D_{rank} = \sum_{f \in features} w_f D_f(pool) = w_{artist} + w_{genre} + \cdots + w_{lyrics}$$

Screenshot from an on-going UX experiment to understand user playlist preferences

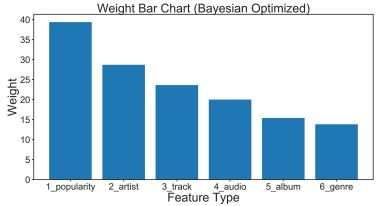
4. Evaluation

Given:
- Incomplete playlist of length **k** (10, 25, 50, 100). - Held out tracks **G**.
- Set **R** of 500 recommend tracks.

Goal: **R** should be "close to **G**" in order.

Metrics:
- r-precision (overlap between sets)
- NDCG (quantify ranking)

k	rprecision_tracks	rprecision_artist	NDCG_tracks	NDCG_artist
10	0.26	0.16	0.13	0.41
25	0.19	0.16	0.13	0.36
50	0.15	0.14	0.15	0.36
100	0.11	0.12	0.17	0.43



5. Future Directions

- Incorporating UX data to understand playlist creation habits and ground truths
- Generating playlist suggestions for new users

* References added in detail in project report

School of Engineering and Applied Sciences, Institute of Applied Computational Science
Harvard University