RISE RESEARCH INNOVATION SCHOLARSHIP ENTREPRENEURSHIP

Abstract ID#: 3115
Category: Interdisciplinary Topics, Centers and Institutes
Undergraduate

Northeastern University

Autoencoded and Graph-Theoretic Music Analyses for Favorable Recommendations

Anirudh Kamath

Opportunity

- Current song recommendation algorithms (Spotify, Apple Music) aim to *minimize user input* and still predict what you like based on
  - the songs in your library
  - songs frequently skipped
  - liked/disliked song recommendations
- Results in redundant/repeated recommendations
  - Instead of randomly shuffling the playlist, Spotify uses a *targeted shuffling* algorithm that ranks songs based on listening activity
  - Often limits the shuffle to only frequently listened songs.

Impact

- Combining hops with latent representations from an autoencoder results in a robust recommendation system that provides both ***related AND similar*** songs.

- Using a graph database helps you *channel your music listening journey*
- Instead of endlessly hitting shuffle or rearranging the listening queue, demands can be set such as:
  - *"transition to more upbeat songs"*
  - *"play songs similar to this artist that I may not have been exposed to"*

- This also provides a voice to the newer and up-and-coming artists. Just like how SoundCloud provided these artists a platform to enter the mainstream, by recommending both similar and related tracks, audiophiles can get ***a far more diverse spectrum of songs to listen to.***

Data/Approach

GraphDB

- We can use "hops" in a graph database to connect songs based on common playlists, albums, and artists

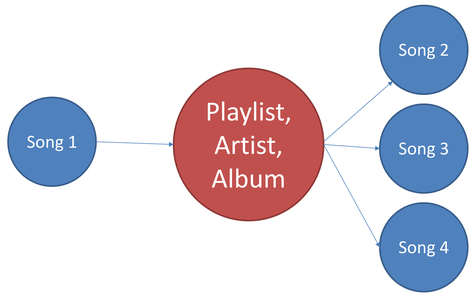
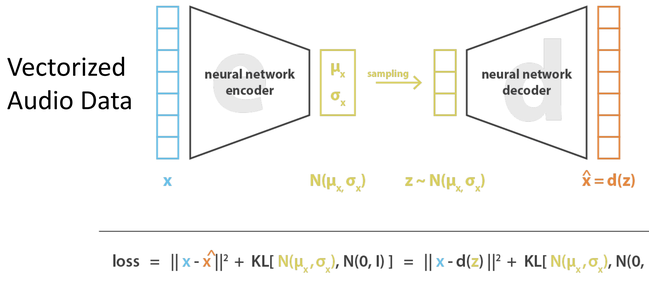


Figure 1 – *Song 1 is 1 "hop" away from songs*

- We can then calculate how *related* two songs are based on how low the number of hops is between two songs

- ***What if two songs aren't connected in the graph?*** Newer artists may not be directly connected to similar artists, but that doesn't mean these artists' shouldn't be recommended! We must measure *similarity*

Variational Autoencoder (VAE)



Source - https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

- Through an autoencoder, we can analyze raw audio segments by pitch, timbre, and loudness and ***analyze how the audio flows through the song***.
- By compressing this data through latent representations in an autoencoder, we get a similarity metric based on the *actual music itself*