PODCAST SEARCH- MUSIC RECOMMENDATION

ANANNYA MATHUR-2019TT10953

ABSTRACT

This project aims to use the Spotify Million Playlist Dataset to understand and implement automated recommendations for the next tracks to listen to. The project aims to implement relatively simpler techniques that are not hugely dependent on computational resources. The project attempts to implement the nearest neighbour method as proposed in the paper "Effective Nearest-Neighbor Music Recommendations"[1] and see if they serve as alternatives to complex learning techniques which also require special-purpose hardware.

CONCEPTS IMPLEMENTED

Nearest-Neighbour, Collaborative Filtering, Information Retrieval

Dataset Description:

Each playlist contains a playlist title, the tracklist (including track IDs and metadata), and other metadata fields (last edit time, number of playlist edits, and more).

The dataset has two parts-

Spotify million playlist dataset (5.39 GB)- Dataset on which models will be trained.

Spotify million playlist dataset challenge(108 MB)- Dataset on which evaluations will be carried out.

Spotify provided a "competitive" dataset consisting of 10,000 playlists, where from each playlist, a certain number of tracks was hidden.

Evaluation Metrics:

R-Precision: Number of retrieved relevant tracks divided by the number of known relevant tracks.

Normalized Discounted Cumulative Gain (NDCG) @500

Recommended Songs Click (Clicks@500): Recommended Songs is a Spotify feature that, given a set of tracks in a playlist, recommends 10 tracks to add to the playlist. The list can be refreshed to produce 10 more tracks. Recommended Songs clicks is the number of refreshes needed before a relevant track is encountered.

APPROACH

For challenge sets that have one or more tracks, we use item-based collaborative filtering. As adapted from the paper mentioned in the reference[1], tf(p,t) can be calculated as $1/(|p|+s)$, where p=playlist, t=track and parameter s is used to ensure that playlists having fewer tracks do not grab very high tfs.

$idf(t)= \log(|P|/|P_t|)$

Ranking of tracks can be done by $rank(t,p)=\sum_{n \text{ in } Np} sim(p,n).1_n(t)$, where $1_n(t)$ is an indicator function that returns 1 if neighbour n contains the track t and 0 otherwise.

For challenge sets that only contain the playlist name but no track to continue recommendation from, we use a string matching algorithm. We find tracks to recommend by collecting playlists that have similar names and then picking tracks based on their frequency in the list. Though the authors propose the usage of fuzzywuzzy library in case no exact match is found, our string match code has only made use of tokenisation of playlist names followed by the application of Porter stemming algorithm.

PRELIMINARY RESULTS:

To make the code more efficient in terms of time and memory, the project considered the top m playlists most recently edited. The parameter m has to be tuned accordingly. Setting m=2000 did fairly well. Parameter s was set at 50 (as proposed in the paper[1]). It is observed that the quality of results degrades if too many neighbours are included. The authors concluded that considering 1000 nearest neighbours(=Parameter K) produced the best results, the project considered 50 neighbours to take into account the memory issues and to speed up the recommendation process. Furthermore, a corpus of pre-computed 500

most popular tracks was maintained (popularity was based on the number of playlists a track appeared in). This was done to address those challenge sets which had less than 500 tracks to recommend.

RESULTS:

Parameter	R-Precision	NDCG	Recommended Song Clicks	Rank
K=50, s=50, m=2000	0.177	0.309	2.702	18
K=50, s=50, m=3000	0.18046	0.312	2.603	17
K=100, s=50, m=4000	0.183	0.325	2.5	14
K=100, s=50, m=5000	0.184	0.327	2.503	12

References:

[1] Effective Nearest-Neighbor Music Recommendations https://doi.org/10.1145/3267471.3267474

[2] Efficient K-NN for Playlist Continuation | Proceedings of the ACM Recommender Systems Challenge 2018

[3] Automatic Music Playlist Continuation via Neighbor-based Collaborative Filtering and Discriminative Reweighting/Reranking | Proceedings of the ACM Recommender Systems Challenge 2018

[3] Dataset https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge