Two-state Model for Automatic Playlist Continuation at Scale

- - http://www.cs.toronto.edu/~mvolkovs/recsys2018_challenge.pdf

ABSTRACT

Automatic playlist continuation is a prominent[1] problem in music recommendation. Significant portion of music consumption is now done online htrought playlists and playlist-like online radio stations. Manually compiling playlists for consumers is a highly time consuming task that is difficult to do at scale given the diversity of tastes and the large amount of musical content available. Consequently, automated playlist continuation has received increasing attention recently. the 2018 ACM RecSys Challenge is dedicated to evaluating and advancing current state-of-the-art in automated playlist continuation using a large scale dataset released by Spotify. In this paper we present our approach to this challenge. We use a two-state model where the first stage is optimized for fast retrieval, and the second stage re-re-ranks retrieved candidates maximizing the accuracy at the top of the recommended list. Our team vl6 achieved 1'st place in both main and creative tracks out of over 100 teams.

KEYWORDS

Playlist Continuation, Collaborative Filtering, Convolution Neural Network Gradient Boosting

Playlist, Continuation,, Collaborative, Filtering,, Convolution, Neural, Network, Gradient, Boosting

1. INTRODUCTION

As music consumption shifts towards online playlists, automated playlist continuation is becoming an increasingly more important problem in music recommendation. While significant progress has been made in recent years, majority of published approaches are evaluated on proprietary datasets making open collaboration and benchmarking difficult. The 2018 ACM RecSys Challenge aims to bridge this gap by conducting standardized evaluation of playlist continuation models. At the core of this challenge is the Million Playlist Dataset(MPD) released by Spotify. MPD is the largest publicly available dataset of it's kind with 1M palylists and over 2.2M songs. The challenge task is to create a playlist continuation model that is able to accurately recommend next songs for each playlist.

The models are evaluated using a separate set of 10K test playlists for which a subset of songs are withheld[2]. Notably, test playlists vary significantly in length from 0 songs (cold start) to 100 songs. This simulates the production scenario where recommendation model has to perform well during all phases of plyalist creation. To avoid over fitting, test held out songs are not released, and teams are required to submit their recommendations to the evaluation server. Over 100 teams participated in the main track of this challenge and our team vl6 achieved 1'st play in both main and creative tracks.

Our approache is based on a two-stage architecture. The first state focuses on reducing the large

2.2M song search space to a much smaller set of candidates for each playlist. By using a combination of collaborative filtering (CF) and deep learning models we are able to retrieve 20K candidates for each playlist with over 90% recall. Moreover, top-1K songs in these candidate sets already cover close to 60% recall. High recall ensures that most relevant songs are captured in the retrieved set, allowing the subsequent models to only focus on this set without significant loss in accuracy. This, in turn, enables us to apply more sophisticated models in the second stage with minimal impact on run time. In the second stage we develop a pairwise model that directly maps each (playlist, song) pair to a relevance score. By jointly incorporating both playlist and song features into the input, this model can capture pairwise relationships that are difficult to express with traditional CF mothos. The aim for the second stage is to re-rank the candidate songs maximizing the accuracy at the top of the recommended list. In the following sections we describe both stages in detail as well as data partitioning, training and inference procedures.

2. APPROACH

관련 문서

Plugin Backlinks: 아무 것도 없습니다.

[1)]
현저한

[2)]
보류