Supplementary Details: Spotify Capstone Project

Spandan Madan, Ben Sanchez, Mehul Smriti Raje and Tim Lee

1 Model Details

1.1 Pooling Model

For ranking we aim at building an interpretable model that would allow us to get some sense of how important each source of information was in raking a playlist. With this in mind we settled for a linear combination of distances from each type of information source. We assume the embedding are capturing a notion of closeness for playlists, so likely songs for a playlist should be close in one of the the embedded spaces.

Given a set of "seed" songs and a set of pool songs, there are many ways of measuring distance between these sets. We looked at getting the optimal combination of weights and distance measures to optimize ranking metrics (r-precision).

1.2 Ranking Model

Since the weights are continuous and positive, and the choice of distance measures were discrete, and we might possibly in the future want to optimize based on a arbitrary utility function, we decided to utilize Bayesian optimization. Bayesian optimization is a exploration and optimization scheme that aims to minimize the number of function evaluations for values in a parameter space.

One advantage of Bayesian optimization is that it permits noisy samples, so for example, we can compute r-precision over a small subset of playlist. If the results look promising, the optimizer will likely evaluate the objective function again for the parameter combination in order to lower the uncertainty of this estimate. This approach permits Bayesian optimization to optimize over our weights and distance measures in a fast way.

The distances we are considering are: 1) distance of mean vector of seed songs against pool. 2) Minimum song-distance of pairwise distances between seed songs and pool , 3) Minimum song distance of pairwise distances between centroids of seed songs and pool and 4) Minimum song distance of pairwise distances between medoids of seed songs and pool.

Given more time, a tree-based model might be preferred for ranking instead of using distances. The advantage of our approach is that it can be arbitrarily adapted for other models that are not distance based, for example you could learn a ranking model for each feature space and then try to learn the hyper-parameters and weights of each model for a global tanking algorithm. Furthermore given a user or a specific context, weights could be learned for each situation against an arbitrary objective function.

2 User study

We selected two playlist for the million playlist datasets that had a large amount of followers for different contexts like romance, party, running, work-out, sad, breakup and so on. One of the main purposes behind our study was to collect data on perceived ranking or ordering for specific context of songs. Ranking has been heavily prioritized in the RecSys challenge, we wanted to investigate if ranking is going to be important for user based on their emotional state and on a desired intention for a playlist. To do so, we ask user about their current state and then give them incomplete playlist and ask to rate the emotional intent of the playlist as well as suggest a continuation song for the playlist. Our song pool comes from real playlist, so this allows us to compare against the actual continuation of the playlist.