CSCI-E109a

Spotify Project

Ranking Results

MAP - Mean Average Precision

<http://fastml.com/evaluating-recommender-systems/>

NDGC – Normalized Discounted Cumulative Gain

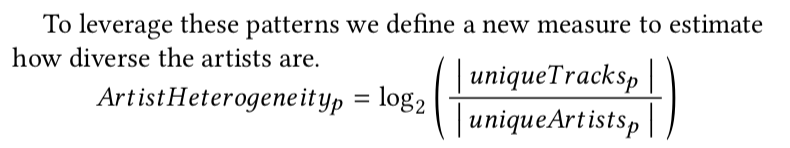
Each item is recommended list gets a relevance score called a “gain”.

Divide each score by the log of the item’s position in the list. “discount” – result is DCG – discounted cumulative gain

Score of each recommended item are summed.

You need to calculate relevance score first.

Artist Heterogeneity – how many unique artists in the playlist



Python Implementation:

<https://gist.github.com/bwhite/3726239>

Playlist(s) need to identified as test or validation playlists.

These are referred to as Ground Truth.

For these playlists, we seed the top song into our model.

The resulting list is compared to the validation

https://recsys-challenge.spotify.com/rules

## Metrics

Submissions will be evaluated using the following metrics. All metrics will be evaluated at both the track level (exact track must match) and the artist level (any track by that artist is a match). In the following, we denote the ground truth set of tracks by G, and the ordered list of recommended tracks by R. The size of a set or list is denoted by |⋅|, and we use from:to-subscripts to index a list. In the case of ties on individual metrics, earlier submissions are ranked higher.

### R-precision

R-precision is the number of retrieved relevant tracks divided by the number of known relevant tracks (i.e., the number of withheld tracks):

R-precision=∣∣G∩R1:|G|∣∣|G|.

The metric is averaged across all playlists in the challenge set. This metric rewards total number of retrieved relevant tracks (regardless of order).

### Normalized discounted cumulative gain (NDCG)

Discounted cumulative gain (DCG) measures the ranking quality of the recommended tracks, increasing when relevant tracks are placed higher in the list. Normalized DCG (NDCG) is determined by calculating the DCG and dividing it by the ideal DCG in which the recommended tracks are perfectly ranked:

DCG=rel1+|R|∑i=2relilog2(i+1).

The ideal DCG or IDCG is, on our case, equal to:

IDCG=1+|G|∑i=21log2(i+1).

If the size of the set intersection of G and R, is empty, then the DCG is equal to 0. The NDCG metric is now calculated as:

NDCG=DCGIDCG.

### Recommended Songs clicks

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. It is calculated as follows:

clicks=⌊argmini{Ri:Ri∈G|}−110⌋

If the metric does not exist (i.e. if there is no relevant track in R), a value of 51 is picked (which is 1 + the maximum number of clicks possible).

### Rank aggregation

Final rankings will be computed by using the [Borda Count](https://en.wikipedia.org/wiki/Borda_count) election strategy. For each of the rankings of *p*participants according to R-precision, NDCG, and Recommended Songs clicks, the top ranked system receives *p*points, the second system receives *p-1*points, and so on. The participant with the most total points wins. In the case of ties, we use top-down comparison: compare the number of 1st place positions between the systems, then 2nd place positions, and so on.