COMP 4106

Project Report

Kanto - Music Recommender System

Hrishi Mukherjee | 100888108

Problem

The introduction of the internet has caused an exponential rise in the access to information. But, within this space of information, exists a lot of noisy and irrelevant content which makes it harder for people to find the content which interests them. One of these content spaces is music. Platforms such as Spotify, SoundCloud, and YouTube provide unlimited access to music, but also face the problem of noisy data.

Motivation

The motivation for this problem comes from observation of people who struggled with finding music matching their preferences. This problem occurred consistently over a period of time, and therefore a solution was required.

Solution

This problem was solved by creating a **Recommender System** for music. This system uses the following Artificial Intelligence techniques:

1. Collaborative Filtering
2. Clustering
3. Classification

The system does pre-processing before the recommendations can be generated. The following outlines the pre-processing procedure:

1. User Query Generation
2. User Preference Computation
3. Preference Based Clustering
4. Cluster Classification

Firstly, an arbitrary number of queries are generated for each user to mimic real search activity. These queries are generated randomly using **real data** crawled from the internet.

Secondly, the query data is used to create a profile for each user. This profile is a representation of the user's music genre preferences which are rated on a scale of 0-100 (0 being not preferred at all, and 100 being completely preferred.) The ratings are generated by analyzing the frequency of the genre in the combined search results of all the user's queries. Using this information, a rating to each genre is given. By the end of this step, each user has a profile containing a rating for each genre.

Thirdly, all the users are clustered based on their profiles. K-Means was chosen as the algorithm to perform clustering. Before K-Means is executed, each user profile is mapped to a vector with each dimension representing a genre rating. These vectors are then fed into K-Means which clusters them in the n-dimensional space. The clustering algorithm is able to produce any amount of clusters. For this solution, four clusters was found to be the optimal number of clusters (least error from center of cluster). By the end of this step, all the users have been clustered with their closest neighbors.

Finally, the clusters of users are classified by preferences. To accomplish this, The average preferences of each cluster are computed. A crucial step after this is to determine a threshold for each preference. This threshold is determined by comparing the average of each preference between clusters and determining a value to classify the clusters evenly. Once a set of thresholds is generated wrt to the features, each cluster is classified. A cluster is classified as falling into a specific class if it's average preference is greater than or equal to the threshold of that feature.

Once the pre-processing procedure is over, we have four clusters of users with each cluster representing a set of preferences of music genres. The recommendations for each user are generated based on the cluster they are classified into. For example, if a user falls into a cluster which is classified as preferring the music genres rock and classical, then that user will be recommended with the a set of rock and classical songs.

Design Choices

The following outlines the design choices used for the solution:

1. Genres were determined based off of the set of real songs crawled from the web.
    a. Electronic Dance Music
    b. Hip Hop
    c. R&B
    d. Pop
    e. Rock
    f. Reggae
    g. Metal
    h. Country
2. User queries were extracted from the set of real songs. Exactly 50 queries per user.
3. The number of users was arbitrarily chosen as 1000.

4. The recommendations were derived from a set of ~7000 real songs.
5. The thresholds for each feature are computed such that the clusters have an even distribution of genre preferences.

Results

The song recommendations being generated for the users were accurately representing the genre preferences of the cluster they belonged to. Additionally, these recommendations were in line with their personal preferences as well. Therefore, the recommendation was a success.

Due to the nature of the data from which the user queries were generated, they tended to incline towards rock music. As a result, the users' rock preferences would tend to be a higher rating compared to other genres on average. This was a flaw in the random generation of queries which could be improved if a data source was used which was uniformly distributed over genres.

Enhancements

The recommendation system can be enhanced by applying more advanced techniques of Collaborative Filtering such as the Nearest Neighbour algorithm. This enhancement would require an enhanced data set which would contain feedback/reviews from the user for the songs.

References

- (n.d.). Retrieved April 19, 2017, from https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html
- How YouTube's Recommendation Algorithm Works. (n.d.). Retrieved April 19, 2017, from https://www.infoq.com/news/2016/09/How-YouTube-Recommendation-Works
- SoundCloud » Introducing Suggested Tracks. (n.d.). Retrieved April 19, 2017, from https://blog.soundcloud.com/2016/06/22/introducing-suggested-tracks/