Arthur Jacobs, Devan Richter, Danielle Rubin, JX Xu
CMSC 12300 - More Data More Problems

Million Song/Last.fm Dataset Tag Analysis

1. Understanding the Dataset

   We used the Last.fm dataset, which assimilates tags and song similarities of songs included in the Million Song Dataset (280 GB). The dataset contains:
   - 943,347 matched tracks MSD <-> Last.fm
   - 505,216 tracks with at least one tag
   - 584,897 tracks with at least one similar track
   - 522,366 unique tags
   - 8,598,630 (track - tag) pairs
   - 56,506,688 (track - similar track) pairs

   Importantly, while the Last.fm dataset contains similar songs and tags for the various songs included, it does not associate songs with their respective artists.

2. Objective and Hypotheses Tested

   We first create a readily-workable dataset by joining together the Last.fm dataset with the Million Song Dataset in order to access the artist of the songs. From there, we assimilate tags and their counts for each artist, allowing us to analyze and compute similarity scores between artists based on overlapping tags and the relative frequency of said tags for each artist.

   We sought to verify our results via “common sense” checks and seeing similar artists on Wikipedia and other music sites.

3. Algorithms and Big Data Approaches Implemented

   Algorithms: Calculating similarity scores for each pair of artists in the data set constituted the most computationally-intensive task. A simple approach would have been to compute the proportion of matching tags between each pair of artists. However, we decided to further fine-tune our similarity scores by incorporating information on the frequency of each tag within a given artist’s data (i.e. the number of times the tag appears across all of the artist’s songs) and within the entire data set across all artists to determine each tag’s relative rarity and contextual importance. To this end, we implemented the term frequency and inverse document frequency (tf-idf) algorithm, which generates vectors of weighted counts for each artist and tag in the dataset. We then computed the

cosine similarity between each unique pair of tf-idf vectors to obtain the final similarity scores, each ranging from 0 (least similar) to 1 (most similar), for each pair of artists.

Big Data Approaches: Based on the nature of our data and analysis, we decided that MapReduce made the most sense conceptually and in terms of practical implementation. We used MapReduce to generate the "bags of tags" for each artist. Specifically, we consolidated data on the tags for each track and artist from the Last.fm and Million Song data sets and used MapReduce to map each artist's name to a dictionary containing all tags found in that artist's track data and the number of songs in which each tag appeared. This output was then passed to the similarity scoring algorithm described above.

4. Runtime
   While the data processing and MapReduce steps were achievable (though slow) using a single machine, the runtime for the similarity scoring algorithm quickly became prohibitive and, after several hours, often generated memory errors when run on a single machine, thus demonstrating the need to utilize cloud computing resources for this project.

5. Challenges Faced
   We encountered numerous difficulties in the process of unpacking, analyzing, and working through the dataset. Below, we outline a few of the more prominent issues we faced:
   a. We originally wanted to work on the reduced redundancy USENET corpus (2005-2011), which ultimately proved infeasible due to inconsistent formatting and organization across the articles in the corpus. Although we made substantial attempts to tackle the dataset via MRJob and MPI functions, we ultimately made little-to-no headway on what proved to be a largely indecipherable dataset. As such, we altered our project to focus on the Million Song Dataset instead, which was a substantial setback.
   b. On our initial approaches to the dataset, we attempted to read in the raw data, which consisted of a JSON file for each individual track. However, this was a very computationally expensive task that resulted in much unhappiness from the sheer number of files.
   c. A major struggle was attempting to make the different code blocks of the project communicate. For instance, we reformatted our MapReduce output several times to create a data structure that was more easily accessible and usable as a function parameter for our td-idf vectors. This

misalignment led us to employ various file types and file loading/reading commands.

6. Outcomes

The validity of our results was somewhat difficult to test due to the qualitative nature of the data itself (music). However, we can report that there was a fairly wide range of scores, and many of the pairs of artists with the highest similarity scores were, based on preliminary web searches, actually quite similar in genre, time period, subject matter, etc. This suggests that the cosine similarity scoring metric fairly accurately predicts similar artists. Further testing could be performed in the future to generate a more quantitative assessment by comparing our similarity scores to others computed on this data set, such as the scores provided in the last.fm data set that measure the similarity of pairs of individual tracks.