Analyzing Song Tags to Identify Similar Artists

More Data More Problems:

Arthur Jacobs
Devan Richter
Danielle Rubin
JX Xu

Project Overview

- Joined Last.fm dataset on Million Song Dataset (280GB)
  - 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
- Implement Map Reduce to assemble bag of tags for each artist
- Build algorithm to test track similarities based on MR output
- Sample and verify results by hand and Google

MapReduce Implementation

Data Sources:

- Last.fm database
- Million Songs Track_Metadata database
- Tracks with tags text file

Step One:

- Mapper Initialization, Mapper, Combiner, Reducer

Step Two:

- Reducer

Mapper Function:

- Initialization
  - attaches both databases
- strips track_id from text file
- Run query for song title, artist name, and associated tags for given track id
  - Each line contains single tag
- Loops through all the tags, yielding pair of artist name and tag

Combiner:

- Passes through same pair, lowercasing tag value

MapReduce Implementation

Reducer (1):

- Loops through tags for given artist_name key and adds to artist's dictionary where **keys are tag names** and **values are counts**
- Adds artist dictionary to larger dictionary where **keys are artist names** and **values are tag dictionary**
- Yields no values

Reducer (2):

- Applies Algorithm
- Outputs to CSV

Similarity Scoring Algorithm

- Simplest scoring: # shared tags / total # of tags
- Issue: tags have differing frequencies across artists, tracks

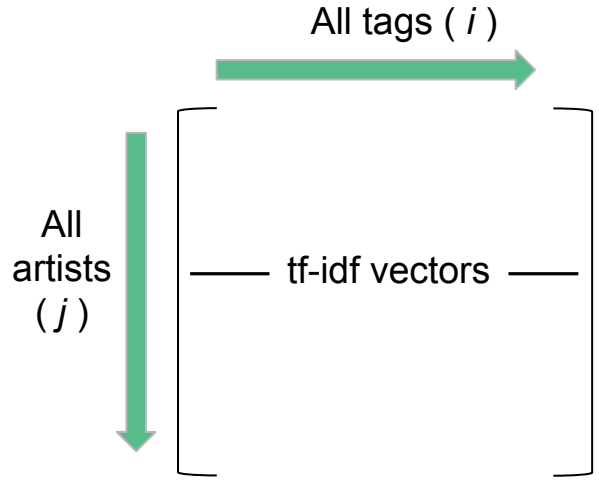
Term frequency-inverse document frequency (tf-idf):

1. Calculate weight of each tag *i* for each artist *j* :

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

2. Calculate similarity scores (cosine similarity) for each pair of tf-idf vectors

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$



Challenges Faced

- Building and segmenting our code to have them communicate effectively and generate output in the correct, usable format
- Testing code proved difficult--errors obscured by VM failures such as insufficient memory
- TF-IDF operations were computationally expensive and enhanced difficulty of error testing

Results

- The result of the cosine similarity algorithm is two artist and a score
- Interpretation: the tags under each artist ‘go the same direction’ when converted to a vector
- Closer to 1 implies that a bag of tags under one artist is similar to a bag of tags under another artist
- Output was clear, but some comparisons are hard to verify
- Overall results are a success for what we set out to accomplish

Artist 1	Artist 2	Cosine Similarity
gob	blue rodeo	0.227285936
gob	duesenjaeger	0.217578149
jamie cullum	tony bennett	0.210367568
tangerine dream	richard souther	0.187808378
casual	snoop dogg	0.182273121
adam ant	frankie goes to hollywood	0.176164771
sonora santanera	andy andy	0.163610256
international noise conspiracy	sofia talvik	0.161088955
extreme noise terror	defecation	0.155783756
sofia talvik	poe	0.149103449
international noise conspiracy	duesenjaeger	0.141997537
junkie xl	motorbass	0.133154575
thomas dutronc	motorbass	0.122078511
cyndi lauper	frankie goes to hollywood	0.121594349
son kite	junkie xl	0.120247935

Questions?