Twitter Music Recommendation Engine

GA Data Science course Fall 2013

Premise

- Create a tool that can give you artist recommendations.
- You enter your twitter handle, it spits back a list of artists whose twitter feeds are most similar to yours.

Overview or Process

- Build library of artist twitter data.
- Extract features.
- Build recommender.
- Interpret Results.
- Tweak features and recommender based on results.

Building my Tweet library

- Used the Echonest API to retrieve the top 1000 artists based on their "hotttnesss" metric.
- Then queried the Twitter API to get the 200 most recent tweets for each artist.
- Slightly over 500 artists had Twitter handles and enough data to be included in the library.

Text processing

- Used sci-kit's provided stop words list. These stop words improved results.
- Sci-kit automatically converts words to lower-case and does some regex processing for you.
- This is an area that needs more focus as I try to further optimize the recommender.

Tf-idf token matrix

- Converted text data to matrix of token counts

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \times \mathrm{idf}(t, D)$$

- Weights a term based on its frequency in a single document and its inverse frequency in the whole corpus.
- My matrix had 160,897 unique tokens.

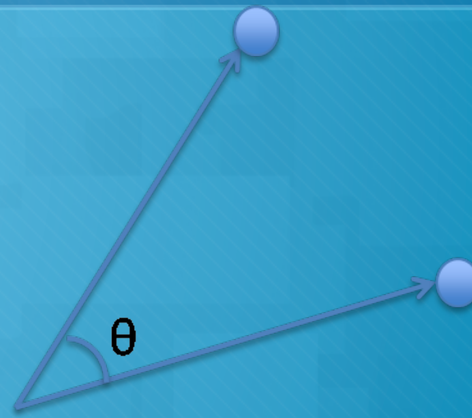
Building the Recommender

- Used NearestNeighbors from sci-kit learn.
- Like KNN, but without the voting process.
- Simply recommends the artists whose vectors are most similar to that of the input handle.

My distance metric

- Intended to use cosine similarity.

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



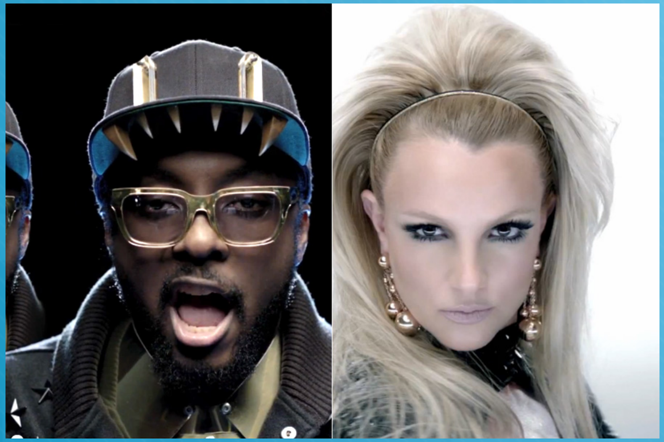
- Ended up using Euclidean distance, as it was faster and gave identical recommendations.

Interpret Results

- This was hard due to unsupervised nature.
- Decided that trying to recommend similar artists could be a good way to qualitatively determine if it was working.

The Good

- Could Identify Artists that collaborated together.



More Good

- Effectively grouped foreign languages together.

The Bad

- A few artists consistently showed up in recommendations.
  - Generic well-formed sentences
  - Location mentions
- Selena Gomez is a pain in the ass.

Next Steps

- Define a systematic approach for evaluating effectiveness of predictions.
- Clustering?
- More feature extraction.
  - NLTK
  - Lemmatisation, n-grams, PCA
  - Really explore the vector space
  - Prioritize mentions and hashtags?

What I learned

- Text data is hard!
- High dimensional vector spaces are hard!
- JSON and API's are nifty!
- Maybe this approach is better at identifying direct links between artists, rather than general recommendations.

Thanks Jamar for the help!