LISTEN, LYRICS TELL THE STORIES

Spring 2017 AC297r: Capstone - Spotify

Lyrics-Based Music Tagging and Recommendation

Presented by:
Heidi Chen, Wenxuan Dai, Xindi Zhao, Yijun Zhou

Partners: Joe Cauteruccio, Aparna Kumar
Advisor: Pavlos Protopapas
TF: Isaac Slavitt
Introduction

❖ Problem Statement and Overview

❖ Description of Data

❖ Modeling Approach

❖ Results and Analysis

❖ Demo

❖ Literature Review and Sources
Right Music at Every Moment

Our solution: Lyrics!
A personalized music enjoying experience with songs that fit users’ moods and lyrics that speak users’ minds.
Description of Data

Original source of music

- Spotify
- Musixmatch

Over 30,000 distinct user-generated tags

- Last.fm
- A-Z Lyrics Database

Lyrics in bag of words format for 5000 top frequency words.

- Song lyrics for over 80,000 songs available
- 14,000 matched with selected tags from Last.fm
What we Explored

Feature Engineering

Bag of Words

Raw Lyrics

Support Vector Machines
Random Forest
Gaussian Naïve Bayes
Long Short Term Memory

Word2Vec
Latent Dirichlet Allocation

--- Unsupervised
--- Supervised
Initial Attempt

Predictors - Lyrics:
Primarily from Million Song Database

Response - Tags:
4 mood tags: Happy, Sad, Energetic, Relax
Baseline Model

Feature Selection
- Remove Stop-Words
- Remove Common Words
- Term Frequency Inverse Document Frequency

Dimensionality Reduction
- Principal Component Analysis (PCA)
- Hashing vectorizer

Predictive Modeling
- Support Vector Machines
- Random Forest
- Gaussian Naive Bayes
## Baseline Model

<table>
<thead>
<tr>
<th>Model</th>
<th>Raw Features</th>
<th>Remove Stop Words</th>
<th>Hashing Vectorizer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian Naive Bayes</td>
<td>21.16%</td>
<td>22.71%</td>
<td>39.04%</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>42.84%</td>
<td>41.93%</td>
<td>37.05%</td>
</tr>
<tr>
<td>Random Forest</td>
<td>36.80%</td>
<td>37.17%</td>
<td><strong>48.80%</strong></td>
</tr>
</tbody>
</table>
Decent baseline, now what?

- Expand the set of tags
  - How to select from the 30,000 tag universe?
  - What are some natural clusters in lyrics?
  - Solution: Latent Dirichlet Allocation

- Improve modeling approach
  - Any more informative predictor? Word2Vec
  - Better approach for language modeling? Long Short-Term Memory
Latent Dirichlet Allocation

Intertopic Distance Map (via multidimensional scaling)

Top-30 Most Relevant Terms for Topic 9 (4.2% of tokens)

- god
- will
- lord
- jesus
- let
- free
- sing
- name
- come
- life
- heaven
- lead
- king
- give
- day
- soul
- holy
- glory
- mercy
- one
- hand
- earth
- world
- stand
- live
- hope
- praise
- hands
- rise
- peace

Marginal topic distribution

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et al (2012)
2. relevance(term w | topic t) = $\lambda$ * p(w | t) + (1 - $\lambda$) * p(w | t)/p(w); see Sievert & Shirley (2014)
Over 30,000 distinct tags

17 selected tags

sad, dark, love, energetic, happy, emotional, funny, religious, political, rain, relax, party, memory, Halloween, Christmas, grunge, and freedom
Final Model

Word2Vec

- Proposed in Mikolov et. al., 2013
- Pre-trained embedding based on Google News dataset (~100 billion words)
- Transform to 300-dimension word vectors.

Long Short-Term Memory (LSTM)

- Proposed in Hochreiter and Schemidhuber, 1997
- Recurrent neural network
- Chain-like structure and memory property good for processing language
Word2Vec Embedding

Embedding Projector

DATA

5 tensors found
Word2Vec 10K

Label by
word

Color by
No color map

 Sphereize data

Load data  Publish

Checkpoint: Demo datasets
Metadata: oss_data/word2vec_10000_200d_

T-SNE  PCA  CUSTOM

X Component #1  Y Component #2  Z Component #3

PCA is approximate.
Total variance described: 8.5%.
Long Short-Term Memory (LSTM)

- 300-dimensional word2vec embedding for each word
- View the first 150 words of each song
- A hidden layer of 100 LSTM memory units
- Final layer of softmax activation mapped to 17 tags

Olah, Christopher. “Understanding LSTM Networks.” http://colah.github.io/posts/2015-08-Understanding-LSTMs/
# Model Results Comparison

<table>
<thead>
<tr>
<th></th>
<th>Random Forest</th>
<th>Long Short-Term Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Train</strong></td>
<td>38.45%</td>
<td>61.12%</td>
</tr>
<tr>
<td><strong>Test</strong></td>
<td>19.13%</td>
<td>38.55%</td>
</tr>
</tbody>
</table>

## Prediction Methodology
- Single Tag: Predict one tag with highest probability
Accuracy Analysis

- High true positive predictability for most tags.
- Love is the most "confounding" tag.
- Highly correlated with emotional, memory, happy, relax and sad.
Alternative Prediction Method

- Top 3: Predict top 3 tags with highest probabilities

- Train: 82.95%
  Test: 60.35%

- High Accuracy: Christmas, love, religious

- Low Accuracy: Energetic, relax
**Auto-tagging:**
For a given song name and artist, we return the top 3 most suitable tags for the song based on the class probabilities predicted from LSTM.

**Song Recommendations:**
With user input keywords, we look for the most related tag and return a list of song recommendations that best fit the tag.
Conclusion

- Word processing techniques such as **Word2Vec** and specialized language models like **LSTM** are effective in improving classification accuracy.

- **Lyrics are useful in labeling songs with the most relevant themes.**

- Complementary to song acoustic features in making playlist recommendations.

**MORE USER GROWTH**

**MORE ENJOYABLE MUSIC EXPERIENCE**

**BETTER USER ENGAGEMENT**

**BUSINESS VALUE FOR ADVERTISERS**
• Certain tags are more appropriate to be predicted by lyrics than others - combining with music acoustic features can make the model more versatile

• Non-mutually exclusive tags - group tags into larger batches or build hierarchical structure within tags with pre-processing


Olah, Christopher. “Understanding LSTM Networks.” http://colah.github.io/posts/2015-08-Understanding-LSTMs/