Combining Audio Content and Social Context for Semantic Music Discovery

Luke Barrington
Douglas Turnbull
Mehrdad Yazdani
Gert Lanckriet

presentation by Andrey Tarasevich
Overview

I. Motivation
II. Information representation
III. Ranking producing algorithms
IV. Experiments
V. Summary
VI. Weaknesses
Motivation

User queries system for a song
“good rock music with nice guitar solos”

1. Rolling Stones
2. Nirvana
3. Metallica
Motivation

What we need to do?

- Represent each song-tag pair with a probabilistic score
- Extract tags from user query
- Rank-order the songs, using relevance score
- Return list of the top scoring songs
Motivation

How we can receive and associate tags for a song?

- Extract information directly from digital representation (frequency)
- Get tags from social source like social networks
Information representation

Social context:

- Social tags
- Web-mined Tags

Audio content:

- Mel frequency cepstral coefficients
- Chroma
Information representation

For each representation the relevance score function $r(s; t)$ is derived

**Sparse:**
- Strength of association between some songs and some tags is missing
- Social context

**Dense:**
- There is always association between song and tag.
- Audio content
Representation of social context

Annotation vector: \( V_s = (v_1, v_2, ..., v_N) \)

- Each element – relative strength of association between song and tag
- Can have *noise*

Mostly sparse because of 2 reasons:
- Tag is not relevant
- Nobody annotated the song with tag
Social tags

Last.fm

Allows user to contribute social tags through their audio player interfaces

By September of 2008:

• 20 million users
• 3.8 million items was annotated over 50 million times
• 1.2 million unique free-text tags
Social tags

For each song in the dataset collect 2 lists of social tag from the Last.fm

First list:
• Consist of relations between song and set of tags

Second list:
• Association between artist and tags
• Aggregates the tag scores for all the songs by that artist
Social tags

Sum tag scores on the artist list and song list plus tag score for any synonyms or wildcard matches tag on either list

\[ \text{Relevance score} \quad r_{social}(s; t) \]
Social tags

“down tempo”

“slow beat”

“hard rock”

“rock”

“electric guitar”
Web-mined tags

1. Collect the document corpus
   1. Query a search engine with the song title, artist name and album title
   2. Retain mapping of documents $M$, such that $M_{s,d} = 1$ if song $s$ was found in the document.

2. Tag songs
   1. Use $t$ as a query string to find the set of relevant documents $D_t$
   2. For each song sum the relevance weights for all $D_t$
Web-mined tags

Relevance score:

\[ r_{web}(s; t) = \sum_{d \in D_t} M_{s,d} \ast \omega_{d,t} \]

- Relevance weight is a function of the tag and document frequency, number of words in the document, number of documents in a collection. (Match() function of MySQL)
Web-mined tags

- During collecting the document corpus use of *site-specific* queries (site:<music site url>) for following query templates

  "<artist name>" music
  "<artist name>" "<album name>" music review
  "<artist name>" "<song name>" music review
Background
Gaussian Mixture Model

- Convex combination of a n-Gaussian distributions
- Used for a clustering problems

Expectation maximization:
- Algorithm for training GMM
Representation of audio content

*Supervised multiclass labeling*

GMM distribution over an audio feature space for each tag in the vocabulary

Audio track $s$ is represented as a bag of feature vectors

$$X = \{x_1, x_2, ..., x_T\}$$

- $x_i$ - feature vector for a short-time segment
- $T$ – number of segments
Supervised multiclass labeling

1. Use the expectation maximization algorithm to learn GMM distribution

2. Identify a set of example songs

3. Use GMMs to learn the parameters of distribution, that represents the tag
Supervised multiclass labeling

- Given a novel song $s$.
- The set of features $X$ is extracted and the likelihood is evaluated.
- Vector of probabilities is interpreted as the parameters of a multinomial distribution

$$r_{audio}(s; t) \propto p(t|X)$$
Representation of audio content

**Mel Frequency Cepstral Coefficients (MFCC):**
- Represents musical notion of timbre
- “color of the music”

**Chroma:**
- Harmonic content representation
- keys, chords
Ranking producing algorithms

We have 4 representation of music information
We have query with a tag

How to produce ranked list?

- Calibrating score averaging (CSA)
- RankBoost
- Kernel combination SVM
Ranking producing algorithms

**Supervised:**
- Use labeled data to learn how best to combine music representation

**Binary judgment labels:**
- for each song-tag pair $l(s;t)$ is denoted.
  - 1 – if pair is relevant
  - 0 – if not relevant
Calibrating score averaging

learn a function $g(\cdot)$ that calibrates scores such that

$$g(r(s;t)) \approx P(t|r(s;t))$$

Allows compare data sources in terms of calibrated posterior probabilities
Calibrating score averaging

Pair-adjacent violators algorithm

- Start with a rank-ordered training set of $N$ songs $s^1, s^2, \ldots s^N$, where $r(s^{i-1}; t) < r(s^i; t)$
- Initialize $g$ such that $g(r(s^i; t)) = l(s^i; t)$
- If data is perfectly ordered, than $g$ is isotonic (non-decreasing)
Calibrating score averaging

Pair-adjacent violation

\[ g(r(s^{i-1}; t)) > g(r(s^i; t)) \]

• To remove violation we update both values with

\[ \frac{g(r(s^{i-1}; t)) + g(r(s^i; t))}{2} \]

• Repeat this, until all violation are eliminated
Calibrating score averaging

There is 7 songs with

\[ r(s; t) = (1, 2, 4, 5, 6, 7, 9) \quad l(s; t) = (0, 1, 0, 1, 1, 0, 1) \]

First, we initialize function \( g(r(s; t)) \)

\[ g(r(s; t)) = l(s; t) = (0, 1, 0, 1, 1, 0, 1) \]

Then we check if the function is isotonic
Calibrating score averaging

here is animated slide with an example

\[(0, 1, 0, 1, 1, 0, 1)\]

\[(0, \frac{1}{2}, \frac{1}{2}, 1, 1, 0, 1)\]

\[(0, \frac{1}{2}, \frac{1}{2}, 1, \frac{1}{2}, \frac{1}{2}, 1)\]

\[(0, \frac{1}{2}, \frac{1}{2}, 1, \frac{3}{4}, \frac{3}{4}, \frac{1}{2}, 1)\]

\[(0, \frac{1}{2}, \frac{1}{2}, \frac{3}{4}, \frac{3}{4}, \frac{5}{8}, \frac{5}{8}, 1)\]
Calibrating score averaging

- Many song-tag scores are missing
- Tag can be actually relevant to the song, but no one annotated song with this tag
- **Estimate** $P(t | r(s; t))$ with $P(t)$

\[
P(t | r(s; t) = 0) = \frac{\#(\text{relevant song with } r(s; t) = 0)}{\#(\text{songs with } r(s; t) = 0)}
\]
RankBoost

Produces strong ranking function $H$ that is a combination of weak ranking functions $h_t$

$$(h_1, h_2, ..., h_n) \Rightarrow H_s = \sum_{i=1}^{n} \alpha_i h_i$$

Each weak function has:

- Representation
- Threshold
- Default value for a missing value
RankBoost

For a given song the weak ranking function is an indicator function, such that

- It outputs 1 if:
  - relevant score > threshold
  - if score is missing and default value if 1
- Otherwise output is 0
RankBoost

1. Initialize $D_t = D$ (weights distribution)
2. Get weak ranking for $h_t$
3. Update weight distributions

$$D_{t+1}(x_0, x_1) = \frac{D_t(x_0, x_1) \exp(\alpha_t (h_t(x_0) - h_t(x_1)))}{Z_t}$$

where $Z_t$ – normalization factor, such that $D_{t+1}$ will be a distribution
RankBoost

During learning:

- The ensemble of weak ranking functions and associated weights is produced

- At each iteration rank loss of a training data is minimized
Kernel combination SVM

Combining sources at the feature level and producing single ranking

- Basically this is linear decision function, that returns
  - positive value, that represents how strong tag is relevant to a song
  - negative value, if tag is not relevant
Experiments

CAL-500 data set

- 500 songs
- 500 unique artists
- 1700 human-generated musical annotations
- Min 3 individuals annotated with 176 tags in vocabulary
Experiments

Assumptions:

- If 80% agree that tag is relevant, then song is considered to be annotated
- Subset of 72 tags is used
- Each tag is annotated with at least 20 songs
- Each tag represents genres, instruments, vocal characteristics, etc.
Experiments

Rank all songs by their relevance

Direct ranking:
- Use relevance score associated with the song-tag pair for a tag

SVM ranking:
- Use SVM and learn decision boundary between “jazz” and “not jazz”
CSA search examples

Top 5 ranked songs for each tag

<table>
<thead>
<tr>
<th>Acoustic Song Texture</th>
<th>Electric Song Texture</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.73 / 0.76</td>
<td>0.76 / 0.73</td>
</tr>
<tr>
<td>Robert Johnson - <em>Sweet Home Chicago</em></td>
<td>Portishead - <em>All Mine</em></td>
</tr>
<tr>
<td>Neil Young - <em>Western Hero</em></td>
<td>Tom Paul - <em>A little part of me</em> (m)</td>
</tr>
<tr>
<td>Cat Power - <em>He War</em> (m)</td>
<td>Spiritualized - <em>Stop Your Crying</em> (m)</td>
</tr>
<tr>
<td>John Lennon - <em>Imagines</em></td>
<td>Muddy Waters - <em>Mannish Boy</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Male Vocals</th>
<th>Distorted Electric Guitar</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.71 / 0.82</td>
<td>0.78 / 0.42</td>
</tr>
<tr>
<td>Bush - <em>Comedown</em></td>
<td>Bush - <em>Comedown</em></td>
</tr>
<tr>
<td>AC/DC - <em>Dirty Deeds Done Dirt Cheap</em></td>
<td>The Smithereens - <em>Behind the Wall of Sleep</em></td>
</tr>
<tr>
<td>Bobby Brown - <em>My Prerogative</em></td>
<td>Adverts - <em>Gary Gilmore’s Eyes</em> (m)</td>
</tr>
<tr>
<td>Nine Inch Nails - <em>Head Like a Hole</em></td>
<td>Sonic Youth - <em>Teen Age Riot</em></td>
</tr>
</tbody>
</table>

Is song is followed by (m), that means misclassification
Prediction of tag relevance

- Chrome representation is the worst
- MFCC takes 60% of tags

<table>
<thead>
<tr>
<th>Representation</th>
<th>Direct Ranking</th>
<th>SVM Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>51</td>
<td>42</td>
</tr>
<tr>
<td>Chroma</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Social Tags</td>
<td>9</td>
<td>21</td>
</tr>
<tr>
<td>Web-Mined Tags</td>
<td>12</td>
<td>9</td>
</tr>
</tbody>
</table>
Table: Single data source results

<table>
<thead>
<tr>
<th>Representation</th>
<th>Direct AUC</th>
<th>Ranking MAP</th>
<th>SVM Ranking AUC</th>
<th>SVM Ranking MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>0.731</td>
<td>0.473</td>
<td>0.722</td>
<td>0.467</td>
</tr>
<tr>
<td>Chroma</td>
<td>0.527</td>
<td>0.299</td>
<td>0.604</td>
<td>0.359</td>
</tr>
<tr>
<td>Social Tags</td>
<td>0.623</td>
<td>0.431</td>
<td>0.708</td>
<td>0.477</td>
</tr>
<tr>
<td>Web-Mined Tags</td>
<td>0.625</td>
<td>0.413</td>
<td>0.699</td>
<td>0.477</td>
</tr>
<tr>
<td>Single Source Oracle (SSO)</td>
<td>0.756</td>
<td>0.530</td>
<td>0.753</td>
<td>0.534</td>
</tr>
</tbody>
</table>

Single Source Oracle – selects best data sources for each tag
Multiple data source results

<table>
<thead>
<tr>
<th>Representation</th>
<th>Direct Ranking</th>
<th>SVM Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>MAP</td>
</tr>
<tr>
<td>Calib. Score Avg. (CSA)</td>
<td>0.763</td>
<td>0.538</td>
</tr>
<tr>
<td>RankBoost (RB)</td>
<td>0.760</td>
<td>0.531</td>
</tr>
<tr>
<td>Kernel Combo (KC)</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

Combination of multiple sources of representation gives significant enhance
Summary

- Each source individually useful for music retrieval (Except Chroma, which is comparable with a random)
- CSA has the best results, but more affected by noise
- Not assigned tags are usually not relevant
- Combination of different music representation allows better calculation of song-tag pair relevant scores
Weaknesses

Small data set
Data set with only 500 songs in compare with any other social network is tiny

It is hard to collect ground truth information for even small set of song-tag pairs
Weaknesses

Reduced number of tags
We deem that over half of tags are redundant or overly subjective

The results of evaluation will be different
Weaknesses

Informal description
Sometimes reader should speculate with definitions

Requires good background in ML
Some algorithms used in this paper are only referenced and not described at all
End