The Goal

To teach computers to appreciate music by finding the emotional qualities of the music.
Recasting the problem

- Use Machine Learning to predict the probability that each of a set of text labels should apply.
How it should work

• Train networks on musical features with accompanying labels

• Then, predict the probability of each label in the label set

• Nirvana’s *Smells Like Teen Spirit* should have high \( p \) for *ANGST-RIDDEN* and *WRY*, and low \( p \) for *CAREFREE*
Add N to (X) -- Party Bag
Boards of Canada -- *Music is Math*
Tom Waits -- *Clap Hands*
How it all works

• Audio features are extracted from MP3 files and used with labels to train network for each label.

• Outputs of those networks become nodes in a Markov Random Field, and belief propagation is run to approximate cultural information to get the final outputs.
Feature extraction

- Each MP3 in the training and test set must have features extracted.
- There has been a fair bit of work in the field.
- Selected work from theses by Pampalk (2001) and Golub (2000).
- These work well, but I could easily use some other method.
Critical band intensity

- Looks at the perceived intensity fluctuation in each *critical band* at frequencies from 1 Hz to 30 Hz.
- Smoothing and other techniques applied, and matrix is unrolled into a vector.
- Robbie Robertson’s *Dance DJ on left*, The Beatle’s *Yesterday* on right.
- Identifies similarity quite well, even in Euclidian space.
Variable-term stats

- Second set of features is a collection of statistics generated using signal processing techniques.
- Successfully used in genre classification.
- Gathers statistics involving intensity range, frequency range, frequency uniformity, etc., over various time scales.
Dimensionality Reduction

- Get 646-dimensional feature vector, high correlated.
- Principal Component Analysis (PCA) is a linear projection that projects to orthogonal dimensions, maximizing the variance.
- Went from 646 audio feature dimensions to 66 principal component dimensions, accounting for 99% of the variance.
Labels

• Each song in training set also has a binary label vector.

• 100 labels, corresponding to genre (ROCK, ELECTRONICA), style (TRIP-HOP, INDIE ROCK) and tone (STYLISH, TENSE).

• Extracted from website that categorizes albums using (much larger) set of keywords.

• Assumption is made that all the labels apply to all the songs on the albums (has interesting consequences...).
• **Moby, *Play***: ELECTRONICA, BROODING, SOPHISTICATED, STYLISH, THEATRICAL, ORGANIC, SENSUAL, PASSIONATE, HOUSE, ALTERNATIVE POP/ROCK, TECHNO, CLUB/DANCE, AMBIENT TECHNO

• **Nirvana, *In Utero***: ROCK, BLEAK, ANGST-RIDDEN, CATHARTIC, REBELLIOUS, WRY, EERIE, VISCERAL, THEATRICAL, TENSE/ANXIOUS, AGGRESSIVE, ACERBIC, RECKLESS, NIHILISTIC, PARANOID, OMINOUS, CONFRONTATIONAL, MENACING, INTENSE, ALTERNATIVE POP, ALTERNATIVE ROCK, GRUNGE
Learning the Labels

- Model is trained for each of the 100 labels.
- Approximately 8000 MP3s used.
- After much experimenting, settled on logistic discriminative network (Neural Net).
- Good for highly nonlinear functions.
- Outputs are probabilities.
Network Topology

features  hidden nodes  label prob.

bias  weighted inputs  activation fn

\[ \sum_{in} \]
However...

- NN doesn’t work equally well for all labels.
- Things like ROCK and ELECTRONICA are easy.
- Things like WHIMSICAL and WINTRY don’t do as well.
Wouldn’t it be nice...

- to let the labels affect each other, leveraging the easy labels to the difficult ones?
- if something is INDIE ROCK and ACERBIC and WRY, it would be ironic if it weren’t IRONIC
- this would allow us to approximate the “cultural” context of the music, by knowing that, when in doubt, a song is more likely to be IRONIC if it’s INDIE ROCK than if it’s HOUSE
- we can make use of the patterns in the labels to do it!
In theory...

- we could use the frequency of label co-occurrence in the training set

- eg if there were 3 labels and we know
  - \( p(L_1=1|L_2=1,L_3=1) \)
  - \( p(L_1=1|L_2=0,L_3=1) \)
  - \( p(L_1=1|L_2=1,L_3=0) \)
  - \( p(L_1=1|L_2=0,L_3=0) \)
  - and observe \( p(L_1=1), p(L_2=1), p(L_3=1) \)

- but for 100 labels, this would require 100 tables with \( 2^{99} \) entries each!
In practice...

- Use only a few labels as parents.
- Using a few most-highly-correlated labels as parents seems to work pretty well.
- This is a loopy graph.
Belief Propagation

- Loops cause problems
- Loopy BP not guaranteed to converge and has unpleasant side effects.
- Use a modified Loopy that terminates after a single pass.
## 5 Highest-Ranked: Neural Net Only

<table>
<thead>
<tr>
<th>Portishead Wandering Star</th>
<th>Leonard Cohen I’m Your Man</th>
<th>Moby Find My Baby</th>
<th>Moby Porcelain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soundtrack</td>
<td>Precious Jazz</td>
<td>Soundtrack House</td>
<td>Manic Gloomy</td>
</tr>
<tr>
<td>Literate</td>
<td>Laid-Back</td>
<td>House Party/Celebratory</td>
<td>Tense Tense</td>
</tr>
<tr>
<td>Precious Organic</td>
<td>Organic Folk-Rock</td>
<td>Precious Hip-Hop</td>
<td>Raucous Raucous</td>
</tr>
<tr>
<td>Organic Druggy</td>
<td></td>
<td></td>
<td>Soothing</td>
</tr>
</tbody>
</table>
## 5 Highest-Ranked: NN + MRF

<table>
<thead>
<tr>
<th>Portishead Wandering Star</th>
<th>Leonard Cohen I’m Your Man</th>
<th>Moby Find My Baby</th>
<th>Moby Porcelain</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Earnest</strong></td>
<td><strong>Autumnal</strong></td>
<td><strong>Electronica</strong></td>
<td><strong>Electronica</strong></td>
</tr>
<tr>
<td><strong>Reflective</strong></td>
<td><strong>Reflective</strong></td>
<td><strong>Playful</strong></td>
<td><strong>Club/Dance</strong></td>
</tr>
<tr>
<td><strong>Wistful</strong></td>
<td><strong>Wistful</strong></td>
<td><strong>Somber</strong></td>
<td><strong>Techno</strong></td>
</tr>
<tr>
<td><strong>Autumnal</strong></td>
<td><strong>Cathartic</strong></td>
<td><strong>Cynical/Sarcastic</strong></td>
<td><strong>Somber</strong></td>
</tr>
<tr>
<td><strong>Adult</strong></td>
<td><strong>Alternative Pop</strong></td>
<td><strong>Aggressive</strong></td>
<td><strong>Calm/Peaceful</strong></td>
</tr>
<tr>
<td><strong>Alternative</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Applications & Future Work

- User study.
- Music Visualization
- Music Exploration/Recommendation
- Improved feature extraction