Intelligent Audio Systems: A review of the foundations and applications of semantic audio analysis and music information retrieval

Jay LeBoeuf
Imagine Research
jay{at}imagine-research.com

July 2008
These lecture notes contain hyperlinks to the CCRMA Wiki.

On these pages, you can find additional supplement the lecture material found in the class - providing extra tutorials, support, references for further reading, or demonstration code snippets for those interested in a given topic.

Click on the symbol on the lower-left corner of a slide to access additional resources.

WIKI REFERENCES...
Review from Day 5

- What are the two parameters that define a RBF SVM?

- What do they (roughly) approximate?

- How did the lab go? Questions on SVM?
One-class SVM

- Binary classifiers rely on positive and negative examples of training data.
- One-class classifiers, however, only rely on positive examples. Great for models where the negative examples are not easily definable. (e.g., a classifier that detects “funky” sounds)
- Parameter: \( \nu \) (“nu”)
One-class SVM

- $\nu$ equals the % of training examples that you are willing to get wrong. (e.g., 10% error rate on training set is $\nu$ of 0.1)
EVALUATION
Our classifier accuracy is 83.4%
Cross-validation

• Say, 10-fold cross validation
• Divide test set into 10 random subsets.
• 1 test set is tested using the classifier trained on the remaining 9.
• We then do test/train on all of the other sets and average the percentages. Helps prevent overfitting.
• Do not optimize too much on cross validation – you can severely overfit. Sanity check with a test set.
Cross-validation
Cross-validation

Fold 1: 76%
Cross-validation

Fold 1: 76%
Fold 2: 80%
Cross-validation

Fold 1: 76%
Fold 2: 80%
Fold 3: 77%
Fold 4: 83%
Fold 5: 72%
Fold 6: 82%
Fold 7: 81%
Fold 8: 71%
Fold 9: 90%
Fold 10: 82%

Mean = 79.4%
Stratified Cross-Validation

- Same as cross-validation, except that the folds are chosen so that they contain equal proportions of labels.
## Evaluation Measures

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>True+</strong></td>
<td>correct</td>
<td>Classifier correctly predicted something in it's list of known positives</td>
</tr>
<tr>
<td><strong>False-</strong></td>
<td>absent</td>
<td>Classifier did not hit, for a known positive result.</td>
</tr>
<tr>
<td><strong>False+</strong></td>
<td>incorrect</td>
<td>Classifier said that something was positive when it's actually negative</td>
</tr>
</tbody>
</table>
Evaluation Measures

“Accuracy”
↑ is good

Precision - “Positive Predictive Value”
↓ = high F+ rate, the classifier is hitting all the time
↑ = low F+ rate, no extraneous hits

Recall – “Missed Hits”
↓ = high F- rate, the classifier is missing good hits
↑ = low F- rate, great at negative discrimination – always returns a negative when it should

F-Measure – a blend of precision and recall (harmonic-weighted mean)
↑
Evaluate Measures

\[ P = \frac{T+}{T+ + F+} \quad [0...1]\]

\[ R = \frac{T+}{T+ + F-} \quad [0...1]\]

\[ F = \frac{2\cdot P\cdot R}{P + R} \quad [0...1]\]
Training and test data

• An overfit model matches every training example (Now it’s “overtrained.”)
• Training Error AKA “Class Loss”
• Generalization
  – The goal is to classify new, unseen data.
  – The goal is NOT to fit the training data perfectly.

• An overfit model will not be well-generalized, and *will* make errors.
• Rule of thumb: favor simple solutions and more “general” solutions.
Training and test data

• Cross-validation

• Training, Validation, and Test set
  – Partition randomly to ensure that relative proportion of files in each category was preserved for each set
    • Weka or Netlab has sampling code

• Warnings:
  – Don’t test (or optimize, at least) with training data
  – Don’t train on test data (no!)
ANALYSIS AND DECISION MAKING
Real-world break

- Toontrack EZ Drummer
  - DrumTracker (Audio -> MIDI transcriber tool)
ANALYSIS AND DECISION MAKING: GMMS
Mixture Models (GMM)

- K-means = hard clusters.
- GMM = soft clusters.
Mixture Models (GMM)

- GMM is good because:
  1. Can approximate any pdf with enough components
  2. EM makes it easy to find components parameters
     - EM - the means and variances adapt to fit the data as well as possible
  3. Compresses data considerably

- Can make softer decisions (decide further downstream given additional information)
GMM Parameters

Input
- Number of components (Gaussians)
  - e.g., 3
- Mixture coefficients (sum = 1)
  - e.g., [0.5 0.2 0.3]
  - “Priors” or “Prior probabilities”
  - Priors are “the original probability that each point came from a given mixture.”
  - “A prior is often the purely subjective assessment of an experienced expert.”
- Initialized centers, means, variances. (optional)

Output
- Component centers/means, variances, and mixture coeff.
- Posterior probabilities
  - “Posterior probabilities are the responsibilities which the Gaussian components have for each of the data points.”

Query
- Similarity via Likelihood or Distance Measure
GMM

• "Pooled covariance" - using a single covariance to describe all clusters (saves on parameter computation)
Fig. 3.1. Spherical covariance mixture model. Sampled data (*dots*), centres (*crosses*) and one standard deviation error bars (*lines*).
4. Spherical covariance mixture model with six components fitted to the
sampled from the full covariance two-component model in Fig. 3.3. Sampled
pts), centres (crosses) and one standard deviation error bars (lines).

From Netlab (p82-83)
Fig. 3.2. Diagonal covariance mixture model. Sampled data (*dots*), centre (*crosses*), covariance axes (*thin lines*) and one standard deviation error bars (*thick lines*).
3. Full covariance mixture model. Sampled data (dots), centres (crosses),
nice axes (thin lines) and one standard deviation error bars (thick lines).
Distance measures between clusters

- The distances between these clusters are computed using the
  - “Centroid distance”
  - Mahalanobis distance
  - Kullback-Leibler Divergence
  - Earth Movers Distance
• Mahalanobis
  – Normalize the distance between the test point(s) and the existing cluster set
    \[ \frac{x - \mu}{\sigma} \]
GMM: EM

- EM is gradient-based – it does not find the global maximum in the general case, unless properly initialized in the general region of interest.
- Log-function is “order-preserving” – maximizing a function vs. maximizing its log gives same results
- Why log? (One idea is to transform an equation’s multiplies into additions, a wonderful property of logs)

\[
\log(x \times y) = \log x + \log y.
\]
Minimization Problems

• Error wants to be $-\infty$, which occurs when Gaussian is fit for each data point. (mean = data point and variance = 0)

• “There are often a large number of local minima which correspond to poor models. Solution is to build models from many different initialization points and take the best model.”
GMM

- Sampling
GMM

• Application:
  – State-of-the-art speech recognition systems
  – estimate up to 30,000 separate GMMs, each with about 32 components. This means that these systems can have up to a million Gaussian components!! All the parameters are estimated from (a lot of) data by the EM algorithm.
PERCEPTUAL INFORMATION:
GENRE
Genre

“Because feature vectors are computed from short segments of audio, an entire song induces a cloud of points in feature space.”

“The cloud can be thought of as samples from a distribution that characterizes the song, and we can model that distribution using statistical techniques. Extending this idea, we can conceive of a distribution in feature space that characterizes the entire repertoire of each artist.”

• **Genre Classification:**
  - Manual : 72% (Perrot/Gjerdigen)
  - Automated (2002) 60% (Tzanetakis)
  - Automated (2005) 82% (Bergstra/Casagrande/Eck)
  - Automated (2007) 76%

*From ISMIR 2007 Music Recommender Tutorial (Lamere & Celma)*
Automatic annotation

- Similarity based on classification

Training Models

Music to be classified

Song 1
Song 2
Song 3
Song 4
Song 5

Database of music 'templates'

Query Song

Trained Models

Query template

Similarity Query

Ordered Results

Song 1
Song 2
Song 3
Song 4
Song 5

C. Decision template 1 (Drum’N’Bass)

D. Decision template 2 (Jungle)

Degree of support

Rock, Classical, Drum’N’Bass, Jungle, Reggae

From ISMIR 2007 Music Recommender Tutorial (Lamere & Celma)
GMM System

- **Separate models for** $p(x|\text{sing})$, $p(x|\text{no sing})$
  - combined via likelihood ratio test

- **How many Gaussians for each?**
  - say 20; depends on data & complexity

- **What kind of covariance?**
  - diagonal (spherical?)
GMM Results

- Raw and smoothed results (Best FA=84.9%):

- MLP has advantage of discriminant training
- Each GMM trains only on data subset  
  → faster to train? (2 x 10 min vs. 20 min)
How?

• One vector
  – High-level features extracted from data
  – Statistics of features extracted from a piece (includes means, weights, etc)
    • Histograms of MFCC features
  – Concatenate features into a single row (encodes time information)
  – MFCC spectral shape
  – “Anchor space” where classifiers are training to represent musically meaningful classifiers. (5 frames of MFCC vectors + deltas)

• Cloud of points
  – Extract audio every $N$ frames
  – K-Means or GMM representing a “cloud of points” for song
    • Clusters: mean, covariance and weight of each cluster = signature
Day 6