

---

# Hidden Markov Classifiers for Music Genres.

---

Igor Karpov (ikarpov@rice.edu)

Rice University

Comp 540 Term Project Fall 2002

---

# The Problem

- Classify digitally sampled music by genres or other categories.
- Categories are defined by “likeness” to other members.
- Solution should be quick, flexible and accurate.

---

# Motivation

- Organize large digital libraries.
- Search for music by melody/sound (second project).
- Understand music better.

---

# Early Considerations.

- Look for common patterns in music.
- The nature of music is sequential.
- Digitally sampled (WAV, MP3, etc.) formats?
  - More readily available and practical.
  - Raw information – harder to deal with.
- Symbolic (MIDI, melody) formats?
  - Less readily available in practical applications.
  - Different order information – some information is lost, some is gained.

---

# Early Considerations.

- Is melodic information enough?
- Consider orchestration, emphasis, etc.
- What are good models for this data?
- Learn from speech recognition, pattern recognition, digital signal processing.

# Previous Work: Folk Music Classification Using Hidden Markov Models.

- Wei Chai and Barry Varcoe and MIT Media Laboratory.
- Input: monophonic symbolic pieces of folk music from Germany, Austria and Ireland.
- Product: 2- and 3 country classifiers using HMMs.
- Results:
  - Hidden state number doesn't matter much (2, 3, 4, 6).
  - Strict left right and left right models are better
  - Interval sequence representation worked best
  - 2 way accuracies of 75%, 77% and 66%, 3- way 63%

---

# Previous Work: Music Classification using Neural Networks

- Paul Scott – last year's term project at Stanford.
- Data: 8 audio CDs in 4 genre categories + 4 audio CDs in 4 artist categories.
- Algorithm: Multi-feature vectors extracted as input to a 20-10-3 feed-forward ANN.
- Product: 4-way genre classifier and 4-way artist classifier.
- Results: genre classification 94.8% accurate, artist classification 92.4% accurate.
- Problematic experimental setup.

---

# Previous Work: Distortion Discriminant Analysis for Audio Fingerprinting.

- Chris Burges et al at Microsoft Research.
- Task: find short audio clips in 5 hours of distorted audio.
- Product: new algorithm for feature extraction (fingerprinting) of audio streams.
- Key: linear neural network does Oriented Principal Component Analysis (OPCA).
- Signal/noise-optimal dimensionality reduction.

---

# Dataset.

- 47-70 songs/genre in MP3 format compressed from 44.1 kHz stereo.
- Converted to Wave-PCM linear encoding 11.025 kHz mono signal.
- Cut 10 evenly spaced ten-second segments per song = 470-700 clips/category.
- 110250 samples per clip.
- 4 categories: rock, techno/trance, classical, Celtic dance.

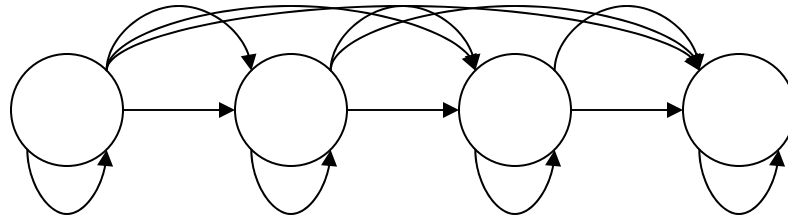
---

# Dataset.

- Easily extracted from real world data
- Contains a lot of information
- Enough for humans to distinguish between genres.

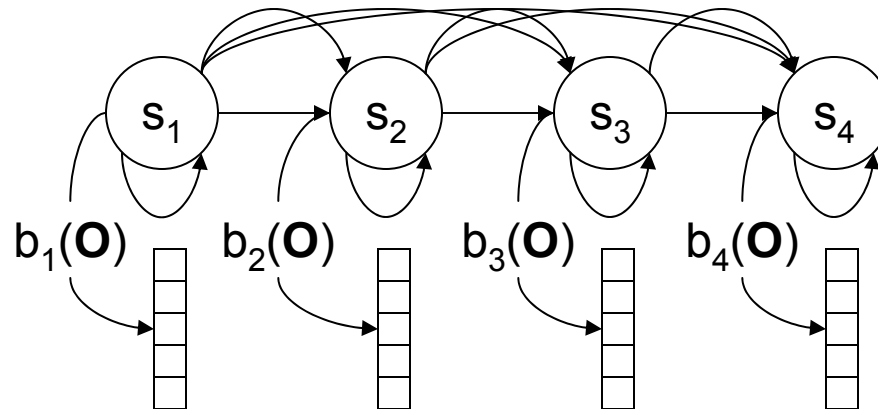
# The Model.

- Continuous Hidden Markov model.
- 3,4 or 5 hidden states.
- Left-to-right architecture



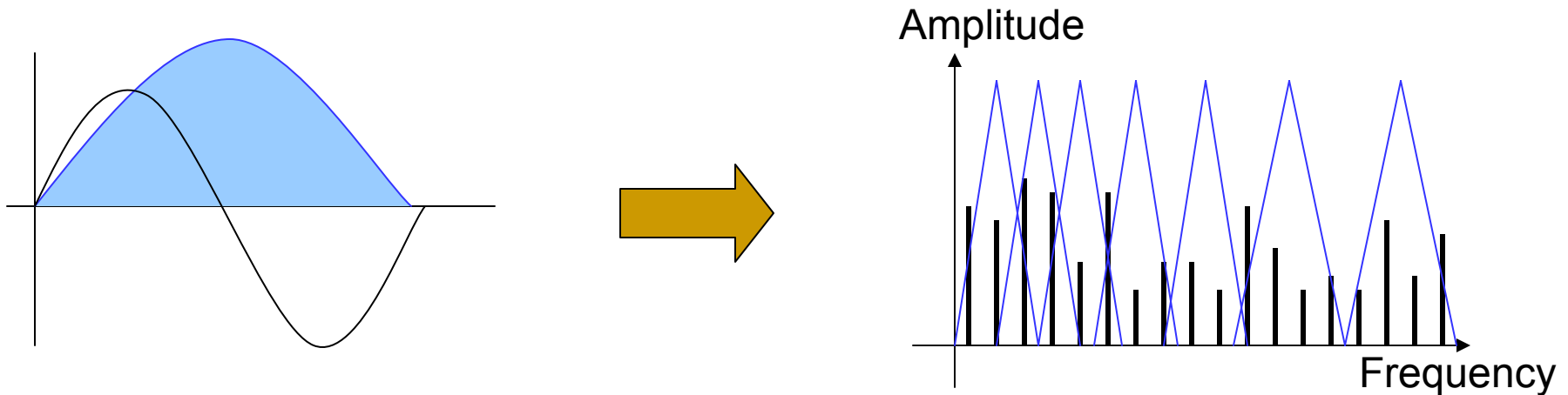
# The Model.

- Each state “outputs” a feature vector with probability distribution  $b_j(\mathbf{O})$ .
  - FFT-based Mel cepstral coefficients.
  - Mel cepstra with delta and acceleration information.
  - Linear prediction cepstral coefficients.
  - (to be implemented) DDA fingerprints.

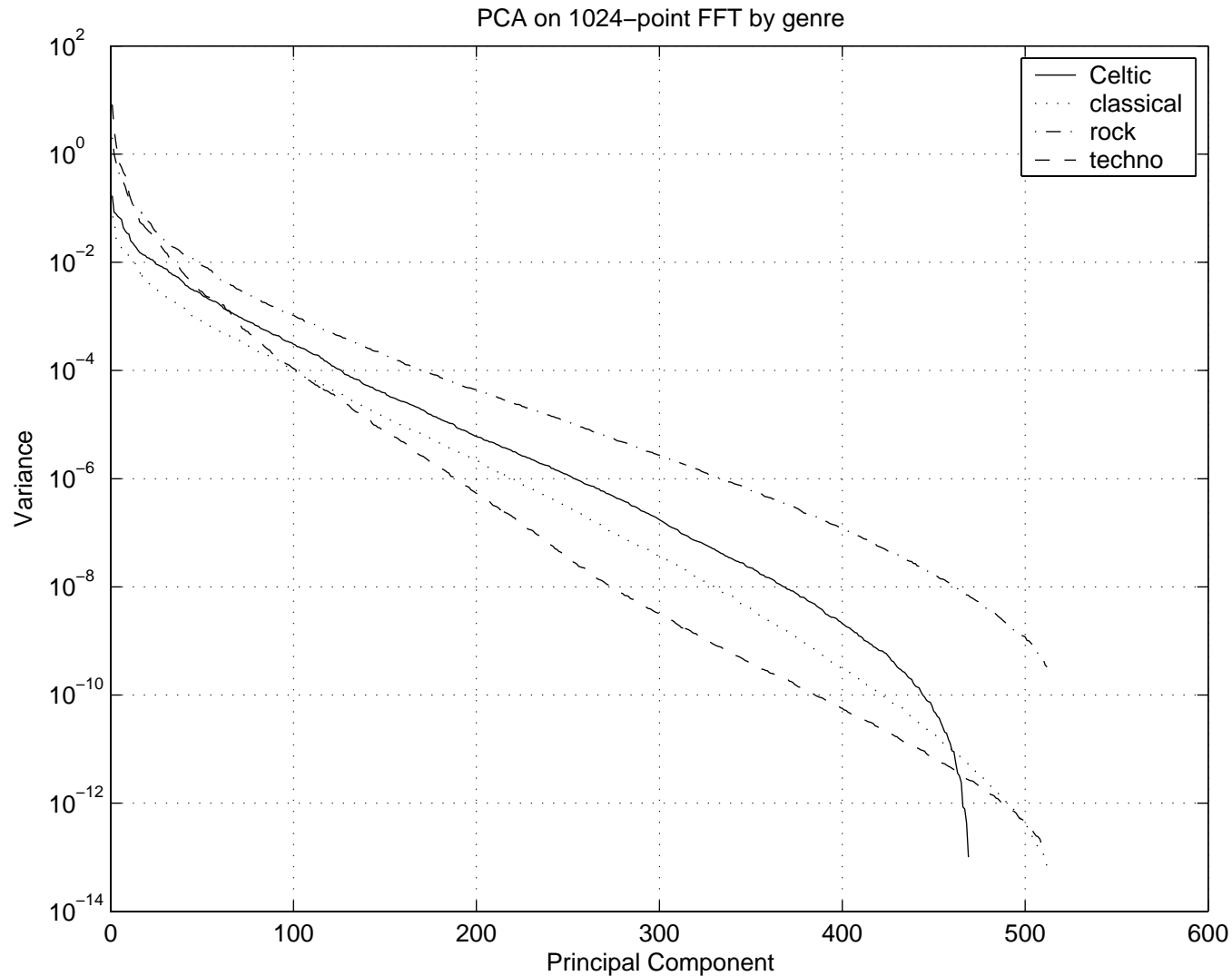


# Feature Extraction: FFT and Mel.

- Pre-emphasize audio signal.
- Multiply by a Hamming window function.
- Take Fourier transform of the window.
- Derive 12 Mel cepstra coefficients from the spectrum. (Models non-linear human audition).



# Features of the Features.



# Feature Extraction: Deltas and Accelerations

- For each Mel coefficient  $C_t$ , append  $\Delta_t = C_t - C_{t-1}$  to the feature vector.
- For each  $\Delta_t$ , append  $a_t = \Delta_t - \Delta_{t-1}$  to the feature vector.
- Enhances the HMM model by adding memory of past states.

# Feature Extraction: LPC.

- Linear Predictive Coding.

- Model the signal as

$$y_{n+1} = w_0 * y_n + w_1 * y_{n-1} + \dots + w_{L-1} * y_{n-L+1} + e_{n+1}$$

- Find the weights that minimize the mean squared error over the window
- 12 weights were used as a feature vector

---

# Feature Extraction: Overall.

- Combine multiple kinds of features into hand-crafted vectors (like Paul Scott).
- Build in prior knowledge about the problem into the learning task.
- (Todo) Use optimizing feature extraction methods like DDA.

# Continuous HMMs.

- Feature vectors are from a continuous domain.
- Two solutions:
  - Discretize the space by finding a representative basis and a distance measure.
  - Use continuous multivariate probability functions.
- Chose to use continuous HMMs.

# Continuous HMMs

- Represent output probability by a mixture of Gaussians.
- Use EM and Baum-Welch reestimation to get the Gaussian parameters and mixture coefficients.
- What should M be? Many parameters vs. expressive power. M=1 worked well.

$$b_j(\mathbf{O}) = \sum_{m=1}^M c_{jm} N(\mathbf{O}, \mu, \mathbf{U}_{jm})$$

---

# The Platform.

- HTK library, originally developed for speech recognition at Microsoft, now at Cambridge.
  - Feature extraction tools.
  - Continuous HMMs.
  - Baum-Welch and Viterbi algorithms.
  - Optimized for performance.
- Worked well – the only thing I had to write were scripts, models and data converters.

---

# The Algorithm.

- One HMM “M” for each category
- Use Baum-Welch reestimation for 20 steps (or until convergence) to obtain M that maximizes  $\log P(O_{\text{training}}|M)$ .
- Use Viterbi algorithm to obtain  $\log P(O_{\text{test}}|M)$  for each category.
- Pick the greatest.

---

# Problems

- Celtic data set is similar to classical and rock and smaller than the other three.
- Failed to find MIDI copies of the dataset.
- Viterbi training did not have enough examples for the number of parameters even with 3-state HMM – undetermined parameters – Had to use Baum-Welch.
- Memory-intensive training – had to use dedicated Ultra-10s in parallel.

# Results: 4-way by 12 MFCC.

- 70%/30% training/test split
- 470 clips/category
- 15 cross-validation trials per experiment
- Top: 12 Mel Cepstral Coefficients
- Bottom: delta and acceleration coefficients added.
- 4 hidden states.

---

<b>4 state HMM</b>	<i>Tech.</i>	<i>Class.</i>	<i>Rock</i>	<i>Celt.</i>
techno	<b>88.2</b>	7.5	2.7	1.5
classical	9.1	<b>74.3</b>	1.7	14.8
rock	4.1	1.6	<b>82.4</b>	11.9
Celtic	3.6	13.0	12.2	<b>71.1</b>

---

<b>4 state HMM</b>	<i>Tech.</i>	<i>Class.</i>	<i>Rock</i>	<i>Celt.</i>
techno	<b>92.4</b>	5.4	1.9	0.3
classical	3.0	<b>88.1</b>	2.4	6.4
rock	2.9	2.1	<b>84.7</b>	10.3
Celtic	1.5	12.1	14.0	<b>72.4</b>

---

# Results: 4-way by 12 LPC.

- 12 LPC cepstra of 14-order LPC
- Same experimental conditions as before.

---

<b>4 state HMM</b>	<i>Tech.</i>	<i>Class.</i>	<i>Rock</i>	<i>Celt.</i>
techno	<b>85.1</b>	8.9	3.5	2.5
classical	10.4	<b>74.6</b>	1.8	13.1
rock	2.2	2.1	<b>85.3</b>	10.4
Celtic	2.4	13.1	12.5	<b>71.9</b>

---

# Results: Varying Hidden State Number.

- 660 clips per genre
- 12 Mel cepstral coefficients with deltas and accelerations (36 total features)

---

<b>3 state HMM</b>	<i>Tech.</i>	<i>Class.</i>	<i>Rock</i>
techno	<b>91.3</b>	7.1	1.5
classical	3.9	<b>93.7</b>	2.4
rock	2.7	4.0	<b>93.3</b>

---

---

<b>4 state HMM</b>	<i>Tech.</i>	<i>Class.</i>	<i>Rock</i>
techno	<b>93.4</b>	4.4	2.2
classical	3.9	<b>94.2</b>	1.9
rock	2.4	3.3	<b>94.3</b>

---

---

<b>5 state HMM</b>	<i>Tech.</i>	<i>Class.</i>	<i>Rock</i>
techno	<b>93.0</b>	5.0	2.0
classical	5.0	<b>93.6</b>	1.4
rock	2.4	3.2	<b>94.4</b>

---

# Results: Generalization

- Verify that we are generalizing across songs.
- An entire song must be either all training or all test.
- Top: random selection (15 cross-validated)
- Bottom: constrained selection (15 c.v.)

---

<b>5 state HMM</b>	<i>Tech.</i>	<i>Class.</i>	<i>Rock</i>
techno	<b>93.0</b>	5.0	2.0
classical	5.0	<b>93.6</b>	1.4
rock	2.4	3.2	<b>94.4</b>

---

<b>5 state HMM</b>	<i>Tech.</i>	<i>Class.</i>	<i>Rock</i>
techno	<b>93.2</b>	5.1	1.7
classical	7.0	<b>91.0</b>	2.0
rock	2.6	3.4	<b>94.0</b>

---

---

# Conclusions

- HMMs are a viable solution to the problem.
- Number of hidden states does not influence results within limits tested.
- Most information is contained in extracted feature vectors.
- Feature vectors are readily modeled by simple Gaussians.

---

# Conclusions

- Some types of music are harder to recognize than others.
  - Less unique features identifiable by feature extraction (Celtic)
  - Sound like other genres

---

# Conclusions

- Models generalize across songs – not just different segments of the same song.
- Better feature extraction (DDA) is the main factor for improving performance.
- Practically useful tools for sorting MP3s can be easily developed using this technique.