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 supplemental material for lectures - 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.

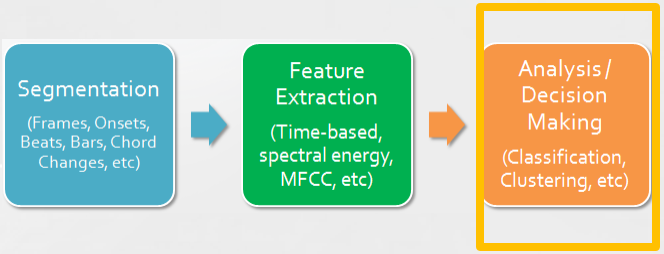
WIKI REFERENCES...

Day 8

Overview of how the rest of the workshop will play out:
Gathering training data / Data prep
HMMs
MPEG-7
Cross validation code
Select "Real-World" Applications
    Feature extraction on your host
Thurs: PM
    Perry Cook visit and real-time code demos

Projects
    Individual review
    Work on your designs and process
    Dive in with the skeletal outline of them

Summary
Resources / References



ANALYSIS AND DECISION MAKING: BUILDING AND EVALUATING A CLASSIFIER

Building a classifier

- Define classes (through training examples)
- Define features
- Define decision algorithm (parameters tuned through training data)
- Evaluate performance (error rate)

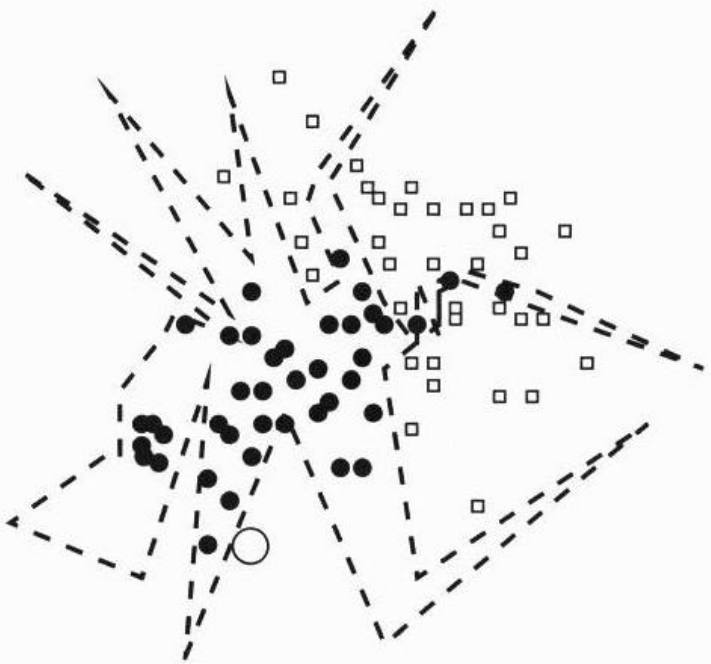


Fig. 2.13. Supervised classification into two classes with 2-dimensional data. In the training set (X, Y), data with label $\mathsf{y} = -1$ are represented with dots, whereas data with label $\mathsf{y} = 1$ are represented with squares. The dotted line is a classification function F such that $\mathsf{R}^{\mathrm{emp}}_{(\mathsf{X},\mathsf{Y})}[\mathsf{F}] = 0$. Though it achieves zero empirical risk, F is not a good classification function, as it makes an error for a new datum which is not in the training set (circle at the bottom, with the true label $\mathsf{y} = -1$).

Building classifier (practical)

- Let's say your building an Electric Guitar instrument classifier.
- It returns tambourines.
- Is this wrong?

When to add negative examples…

DATA PREPARATION

Gathering training data

- Hand-annotate
- One technique is to use existing ground-truth MIDI files to synthesize audio.
- Contact researchers regarding their annotations
- Creative Commons, research collections (RWC)

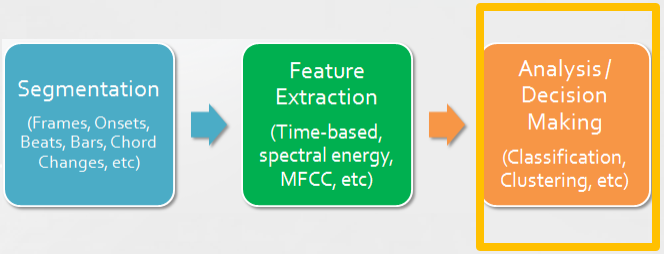
Data preparation

- Examine your data at every chance. (means, max, min, std, etc)
- Always try to visualize data to see patterns and see if it makes. Incredible sanity check.
- Sonify when possible
- Look and eliminate noisy data
- Data preparation
  - Cleaning
    - Open up and examine
    - Handle missing values
  - Relevance / Feature analysis
    - Remove irrelevant or redundant attributes
  - Data Transformation
    - Generalize or normalize data

An example workflow

- Extract raw features
- Transform data to format of SVM software
- Try a few kernels with default parameter settings
- Test
- But the results are **BAD**.

- What do you think we DID WRONG???



ANALYSIS AND DECISION MAKING

HIDDEN MARKOV MODELS (HMM)

HMM

- So far, we've had memoryless systems
- Temporal pattern recognition
- Could add a simple transition matrix to describe observations
- Popular in speech recognition to model transitions between phonemes. Gigantic databases with labels accumulated over decades.

HMM

- What can we do with them?
  - What is the overall probability of this sequence of observations?

    or
  - What is the overall probability of **this model** generating the given sequence?
  - What is the particular state at any point in a sequence?
  - What is the most likely state path that would explain the observations?
  - Creative: You can use it to generate additional observation sequences given a current starting initial condition…

HMM: Speech

- Speech recognition applications…
  - voice dialing (“call home”)
  - Phone tree or vocal phone dialing (“nine one one”)
  - ASR (speech-text)
    - Also using contextual information

HMM: Examples

- Chord estimation or key estimation # 1
  - HMMs (24-key specific HMMs)
    - States (24) = Chord classes
    - Key is the model whose likelihood is greatest. (Making frame-based decisions. "Panel of experts")
  - Using only the 1 previous chord.
  - Training based on examining MIDI->Audio examples
    - Labeled segments with chord labels, states sequence, and observed key.
  - Testing is to feed in chroma/tonal centroid states and query with Viterbi:
    - "What's the likelihood of this state sequence, evaluating all 24 HMMs?"

      and once we know that…

      "What HMM (key) is it?"

HMM

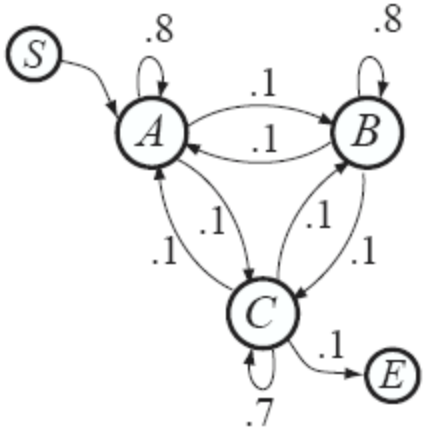
- An important characteristic of Markov models is that the next state depends only on the current state, and not on the history of transitions that lead to the current state.

- **Order** = number of states affecting the choice of the next state

  - **First order process** - decision is made purely on previous state. (probabilistic decision is made) - it's not deterministic)

3 Markov models

- A (first order) Markov model is a finite-state system whose behavior depends only on the current state
- E.g. generative Markov model:



$p(q_{n+1}\|q_n)$		q_{n+1} S	A	B	C	E
q_n	S	0	1	0	0	0
	A	0	.8	.1	.1	0
	B	0	.1	.8	.1	0
	C	0	.1	.1	.7	.1
	E	0	0	0	0	1

S A A A A A A A A B B B B B B B B B C C C C B B B B B B C E

HMM: Nomenclature

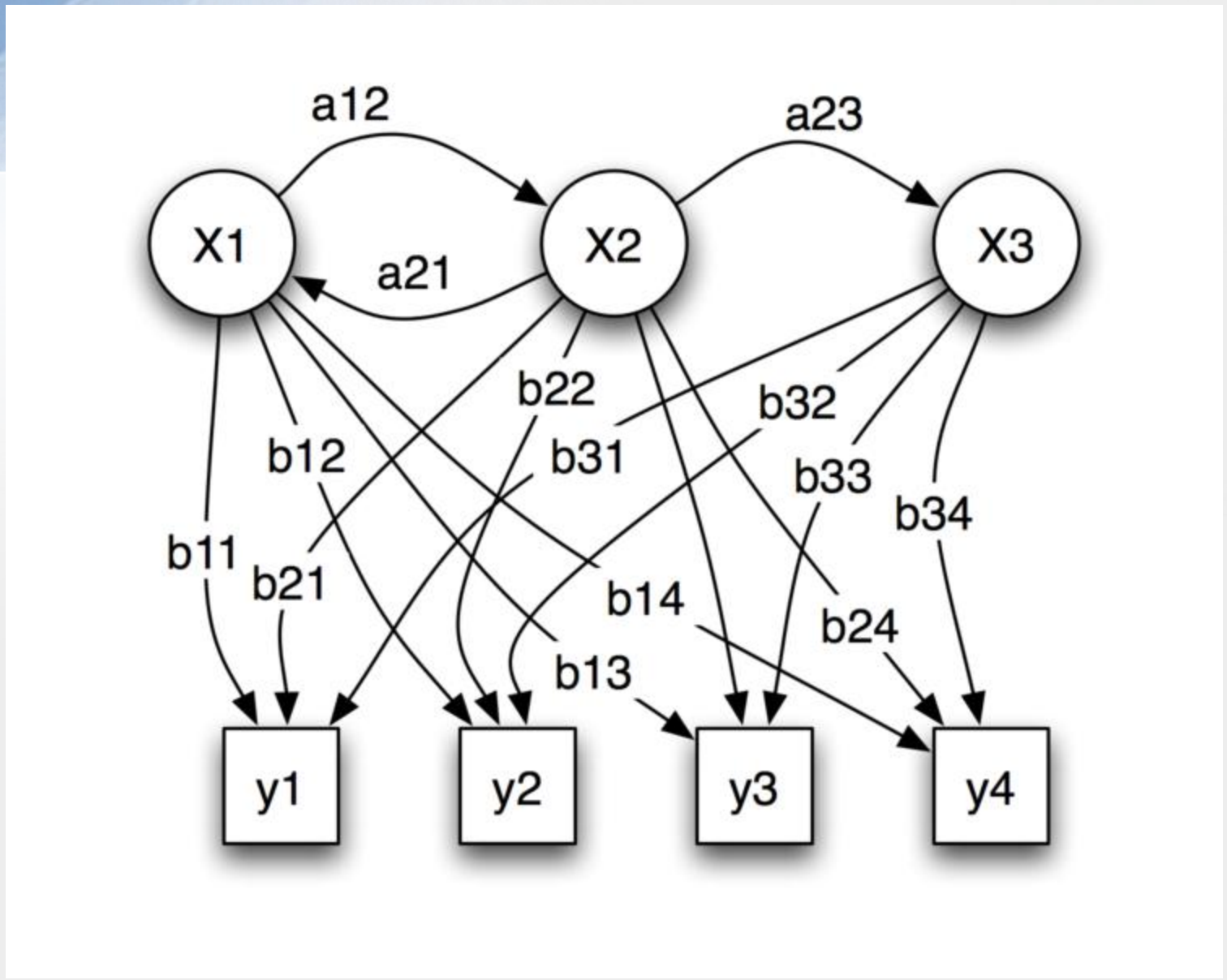
- **State Transition Matrix** is the collection of state transition probabilities. (Can go to other states, or stay on same state)
- "Hidden states": the internal states of the system
- Observations: the visible output of the process
- The number of hidden process may be different than # of observed states

- An HMM is defined by

Pi - vector of initial probabilities

A - state transition matrix

B - confusion matrix


a12
a23
X1
X2
X3
a21
b22
b32
b12
b31
b33
b34
b11
b21
b14
b24
b13
y1
y2
y3
y4

HMM: Speech (a detailed example)

- Word recognition
- Say, 100 words = 100 HMMs
  - Each word has 40 samples of a word

- Build an HMM for each word (determine params)

- For each unknown word, extract features, send into an existing classifier to approximate phonemes, and then query for best observation sequence. Choose model with high likelyhood of achieving that observation sequence.

HMMs : What do we do with them?

1. **Learning**: generating an HMM given a sequence of observations.
2. **Evaluation**: finding probability of an observed sequence given an HMM.
   Decoding: finding the sequence of hidden states that most probably generated an observed sequence.

HMM: Learning

- Find parameters (Pi,A,B) given sequence of observations and the set of hidden states.
  - Find the state transition matrix, observation matrix, and initial vector based on sequence of observations and known hidden set.
  - e.g., speech database where observations are phoneme-sequences of words, and states are the phonemes

- **Forward-backward algorithm** used to find A & B.
  - Calculated by making an initial guess and trying to minimize the error of the system.
- **Baum-Welch**

HMM: Evaluation & Decoding

- Find the most probable sequence of hidden states that generated the observed output sequence.
- **Viterbi algorithm** determines the most probable sequence of hidden states given a sequence of observations and a HMM.
- **Overall best path is the state with the maximal probability and choose it's best path.** (Looks at whole sequence – can tolerate errors midway in sequence, so long as everything else is reasonable)

Viterbi path

- Total probability is over all paths
- Can also solve for "Viterbi state sequence" – the single best path
  - Backtrack from final state to get the best path

HMM: Examples

- Key Estimation # 2 (Nolan)
- Observations = Chord transitions (Any 2 combinations of chords, or "no chord")
    - HMM
        - 24 states (all major and minor keys)
        - **Initialized** to these HMM parameters
            - » Initial state probabilities : $1/24^{th}$, since there's no reason to prefer 1 key over another before hearing music.
            - » State transition: Probabilities of next key in the next time step. (Heavily favors staying in same key or related keys)
            - » Observation probabilities: Reflects our expected and common chord transitions reflecting key-centric behavior.
        - Train with sequence of chord transitions and known keys
    - Decode with Viterbi to find most likely sequence of keys at each time.
    - To find overall, key likelihoods for all time frames are summed- giving overall likehood for each key. Largest likelihood = best key for song.

HMM: More info

- Read for more information:
  - "A tutorial on hidden markov models and selected applications in speech recognition"
    Lawrence Rabiner, Proc. IEEE, 77(2), Feb 1989.
    www.cs.ubc.ca/~murphyk/Bayes/rabiner.pdf

http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/main.html

  - "Hidden markov models for automatic speech recognition: Theory and Application"

> end HMM