Digitizing Speech

Continuous Sound pressure wave → Microphone → Discrete Digital Samples

$s(n)$

Thanks to Bryan Pellem for this slide!
Frame Extraction

- A frame (25 ms wide) extracted every 10 ms

HMMs for Continuous Observations?

- Before: discrete, finite set of observations
- Now: spectral feature vectors are real-valued!
- Solution 1: discretization
- Solution 2: continuous emissions models
  - Gaussians
  - Multivariate Gaussians
  - Mixtures of Multivariate Gaussians
- A state is progressively:
  - Context independent subphone (~3 per phone)
  - Context dependent phone (=triphones)
  - State tying of CD phone
Vector Quantization

- Idea: discretization
  - Map MFCC vectors onto discrete symbols
  - Compute probabilities just by counting
- This is called Vector Quantization or VQ
- Not used for ASR any more; too simple
- Useful to consider as a starting point

Gaussian Emissions

- VQ is insufficient for real ASR
- Instead: Assume the possible values of the observation vectors are normally distributed.
- Represent the observation likelihood function as a Gaussian with mean $\mu_j$ and variance $\sigma_j^2$

$$f(x \mid \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$
Gaussians for Acoustic Modeling

A Gaussian is parameterized by a mean and a variance:

- $P(o|q)$:
  - $P(o|q)$ is highest at mean
  - $P(o|q)$ is low here, very far from mean

Multivariate Gaussians

- Instead of a single mean $\mu$ and variance $\sigma$:
  $$f(x | \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$
- Vector of means $\mu$ and covariance matrix $\Sigma$
  $$f(x | \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} (x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$
- Usually assume diagonal covariance
  - This isn’t very true for FFT features, but is fine for MFCC features
Gaussian Intuitions: Size of $\Sigma$

- $\mu = [0 \ 0] \quad \mu = [0 \ 0] \quad \mu = [0 \ 0]$
- $\Sigma = I \quad \Sigma = 0.6I \quad \Sigma = 2I$
- As $\Sigma$ becomes larger, Gaussian becomes more spread out; as $\Sigma$ becomes smaller, Gaussian more compressed

Text and figures from Andrew Ng’s lecture notes for CS229

Gaussians: Off-Diagonal

- $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}; \quad \Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$
- As we increase the off diagonal entries, more correlation between value of $x$ and value of $y$

Text and figures from Andrew Ng’s lecture notes for CS229
In two dimensions

\[ O_1 \text{ and } O_2 \text{ are uncorrelated – knowing } O_1 \text{ tells you nothing about } O_2 \]

\[ O_1 \text{ and } O_2 \text{ can be uncorrelated without having equal variances} \]

From Chen, Picheny et al lecture slides

In two dimensions

\[ O_1 \text{ and } O_2 \text{ are correlated – knowing } O_1 \text{ tells you something about } O_2 \]

From Chen, Picheny et al lecture slides
But we’re not there yet

- Single Gaussian may do a bad job of modeling distribution in any dimension:

Solution: Mixtures of Gaussians

![Bad News!!!](image)

Mixtures of Gaussians

- M mixtures of Gaussians:

\[
f(x \mid \mu_{jk}, \Sigma_{jk}) = \sum_{k=1}^{M} c_{jk} N(x, \mu_{jk}, \Sigma_{jk})
\]

\[
b_j(o_t) = \sum_{k=1}^{M} c_{jk} N(o_t, \mu_{jk}, \Sigma_{jk})
\]

- For diagonal covariance:

\[
b_j(o_t) = \sum_{k=1}^{M} \frac{c_{jk}}{2\pi^{D/2} \prod_{d=1}^{D} \sigma_{jkd}^2} \exp\left( -\frac{1}{2} \sum_{d=1}^{D} \frac{(x_{jkd} - \mu_{jkd})^2}{\sigma_{jkd}^2} \right)
\]
GMMs

- Summary: each state has a likelihood function parameterized by:
  - M mixture weights
  - M mean vectors of dimensionality D
  - Either
    - M covariance matrices of DxD
  - Or often
    - M diagonal covariance matrices of DxD
      which is equivalent to
    - M variance vectors of dimensionality D

HMMs for Speech

Diagram of a simple HMM for speech with states labeled 'n', 'iy', and 'd', and observation sequence depicted as spectral feature vectors.
Phones Aren’t Homogeneous

Need to Use Subphones

Phone Model

Observation Sequence (spectral feature vectors)
A Word with Subphones

Viterbi Decoding

P(\text{the l of})
ASR Lexicon: Markov Models

Markov Process with Bigrams
Training Mixture Models

- Forced Alignment
  - Computing the “Viterbi path” over the training data is called “forced alignment”
  - We know which word string to assign to each observation sequence.
  - We just don’t know the state sequence.
  - So we constrain the path to go through the correct words
  - And otherwise do normal Viterbi
- Result: state sequence!

Modeling phonetic context

- W iy
- r iy
- m iy
- n iy
“Need” with triphone models

Possible triphones: $50 \times 50 \times 50 = 125,000$

How many triphone types actually occur?

20K word WSJ Task (from Bryan Pellom)
- Word internal models: need 14,300 triphones
- Cross word models: need 54,400 triphones
- But in training data only 22,800 triphones occur!

Need to generalize models.
State Tying / Clustering

- [Young, Odell, Woodland 1994]
- How do we decide which triphones to cluster together?
- Use phonetic features (or 'broad phonetic classes')
  - Stop
  - Nasal
  - Fricative
  - Sibilant
  - Vowel
  - lateral

State Tying

- Creating CD phones:
  - Start with monophone, do EM training
  - Clone Gaussians into triphones
  - Build decision tree and cluster Gaussians
  - Clone and train mixtures (GMMs)
Standard subphone/mixture HMM

Model | Error rate
---|---
HMM Baseline | 25.1%

Our Model

Fully Connected | Standard Model
---|---
Single Gaussians | Single Gaussians
Hierarchical Baum-Welch Training

HMM Baseline 25.1%
5 Split rounds 21.4%

Refinement of the /ih/-phone
Refinement of the /ih/-phone
HMM states per phone

Inference

- State sequence:
  \[ d_1-d_5-d_5-d_4-ae_2-ae_3-ae_0-d_2-d_3-d_7-d_5 \]

- Phone sequence:
  \[ d - d - d - ae - ae - ae - d - d - d - d \]

- Transcription
  \[ d - ae - d \]

Viterbi

Variational

???