

Hidden Markov Model (II)

6. Oct. 2008

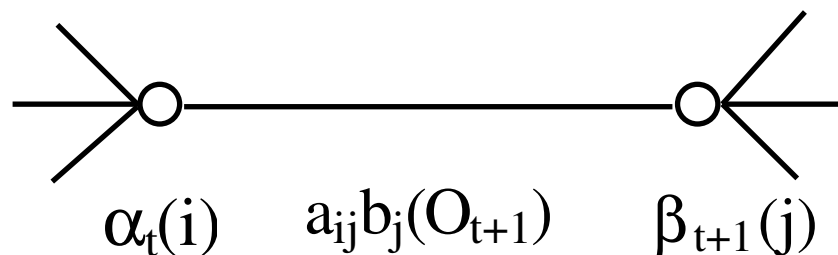
Training HMM Parameters

- Train parameters of HMM
 - Tune λ to maximize $P(O | \lambda)$
 - No efficient algorithm for global optimum
 - Efficient iterative algorithm finds a local optimum
- Viterbi-Training
 - Compute Viterbi-Path using Current Model
 - Reestimate Parameters Using Labels Assigned by Viterbi
- Baum-Welch (Forward-Backward) reestimation
 - Compute probabilities using current model
 - Refine $\lambda \rightarrow \bar{\lambda}$ based on computed values
 - Use α and β from Forward-Backward

Forward-Backward Algorithm

- Probability of transiting from S_i to S_j at time t given O

$$\begin{aligned}\xi_t(i,j) &= P(q_t = S_i, q_{t+1} = S_j \mid O, \lambda) \\ &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O \mid \lambda)}\end{aligned}$$



Baum-Welch Reestimation

$$\bar{a}_{ij} = \frac{\text{expected number of trans from } S_i \text{ to } S_j}{\text{expected number of trans from } S_i}$$

$$= \frac{\sum_{t=1}^T \xi_t(i,j)}{\sum_{t=1}^T \sum_{j=0}^N \xi_t(i,j)}$$

$$\bar{b}_j(k) = \frac{\text{expected number of times in state } j \text{ with symbol } k}{\text{expected number of times in state } j}$$

$$= \frac{\sum_{t: O_t=k} \sum_{i=0}^N \xi_t(i,j)}{\sum_{t=1}^T \sum_{i=0}^N \xi_t(i,j)}$$

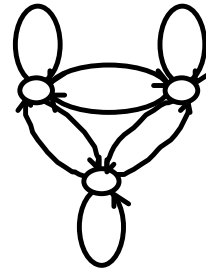
Convergence of FB Algorithm

1. Initialize $\lambda = (A, B)$
2. Compute α , β , and ξ
3. Estimate $\bar{\lambda} = (\bar{A}, \bar{B})$ from ξ
4. Replace λ with $\bar{\lambda}$
5. If not converged go to 2

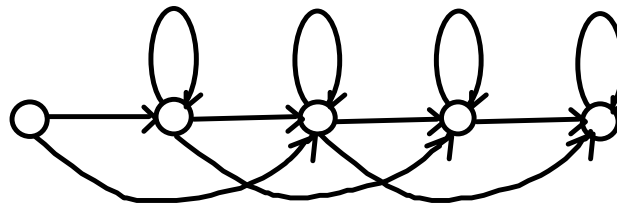
It can be shown that $P(O | \bar{\lambda}) > P(O | \lambda)$ unless $\bar{\lambda} = \lambda$

Model Topologies

Ergodic - Fully connected, each state has transition to every other state

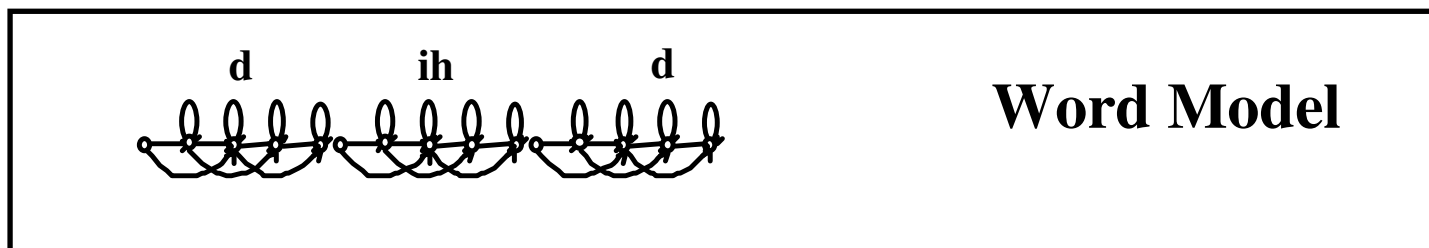
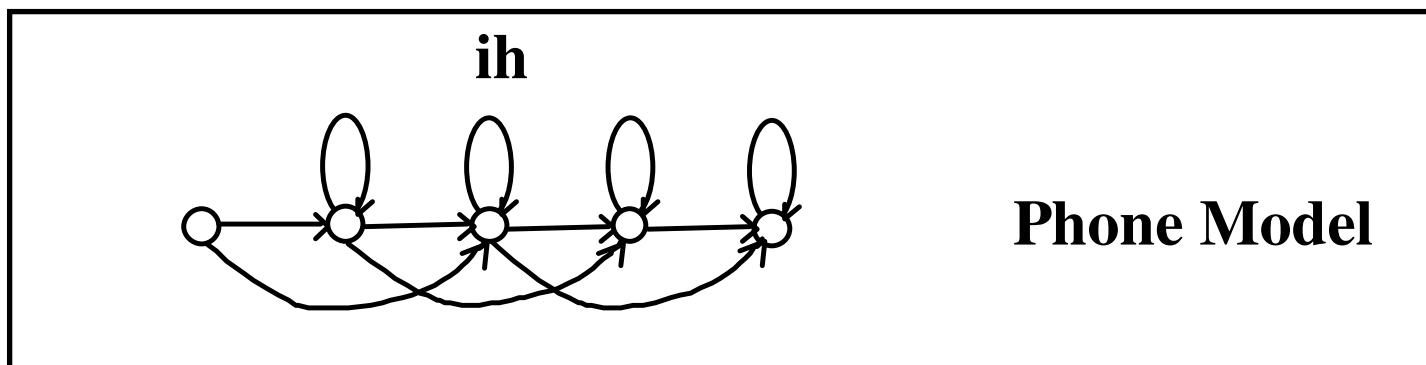


Left-to-Right - Transitions only to states with higher index than current state. Inherently impose temporal order. These most often used for speech.

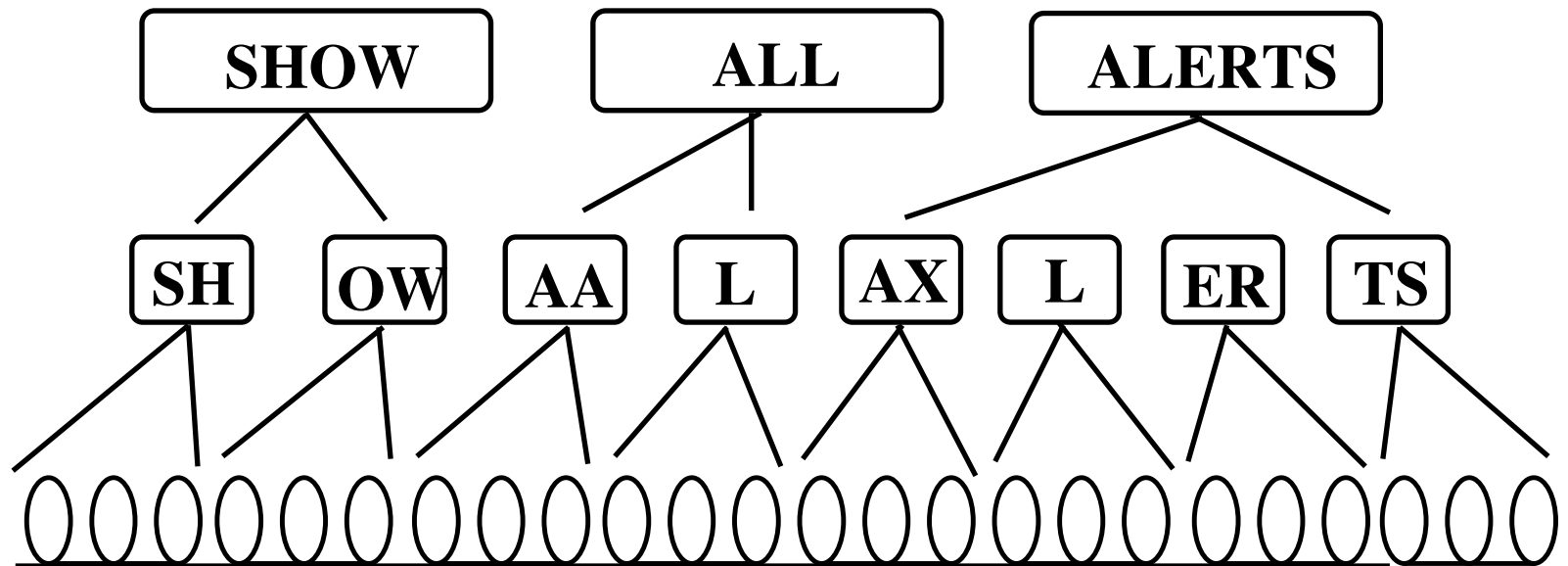


HMMs In Speech Recognition

- Represent speech as a sequence of observations
- Use HMM to model some unit of speech (phone, word)
- Concatenate units into larger units

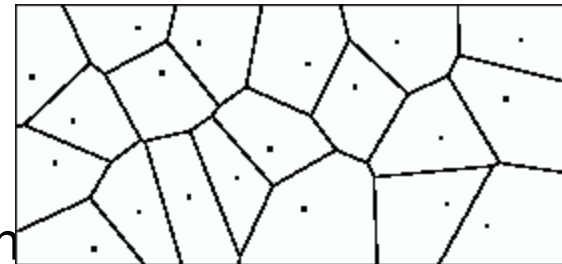


Forward-Backward Training for Continuous Speech



Discrete HMM's Vector Quantization

- Want to create a discrete set of areas that we can label
- choose a number of prototypical reference vectors
- the set of points to which a reference vector is the closest is called the vector's **Voronoi region**
- every Voronoi region is convex
- Give each Voronoi region a label, „code“
- Set of codes is the „code book“
- During Classification, assign label of the region new sample falls
- Sequence of labels represents „Observations“ of a Discrete HMM



Acoustic Modeling

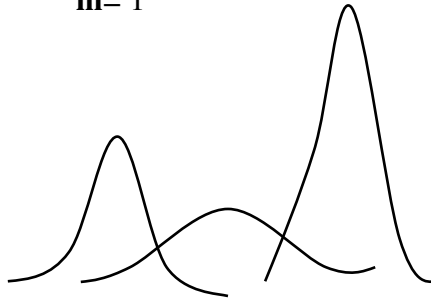
- How to Model Emission Probabilities
 - Discrete
 - Continuous
 - Mixture Gaussians
 - Neural Nets
 - Semi-Continuous, Tied Mixtures
- The Problem of Context
 - The Markov Assumption is really not Good!
 - Context-Dependent Phones
 - Tri-Phones
 - Poly-Phones

Acoustic Modeling

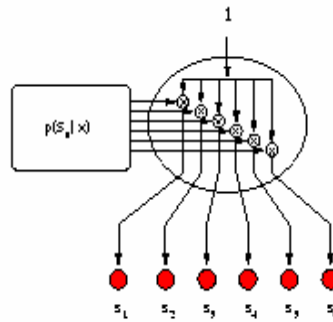
Emission Probabilities can be Estimated by Alternate Methods:

Mixture of Gaussians Networks

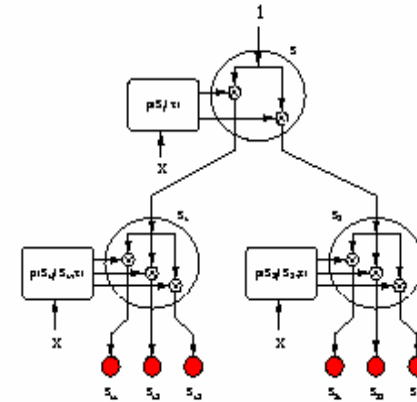
$$b_j(x) = \sum_{m=1}^M c_{jm} N[x, \mu_{jm}, U_{jm}]$$



Neural Networks

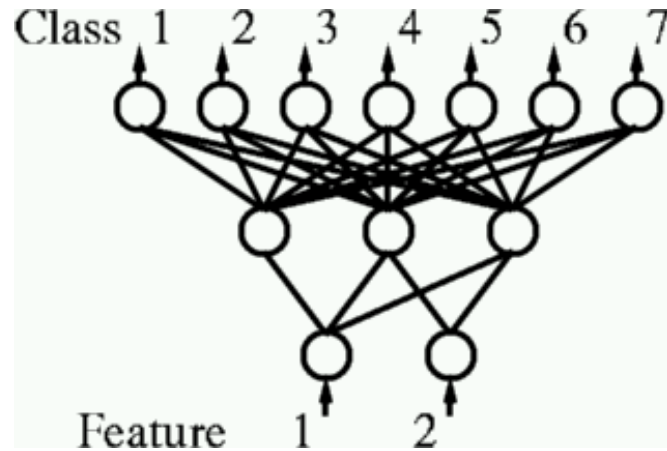


Hierarchies of Neural



Neural Net Approaches to Pattern Classification

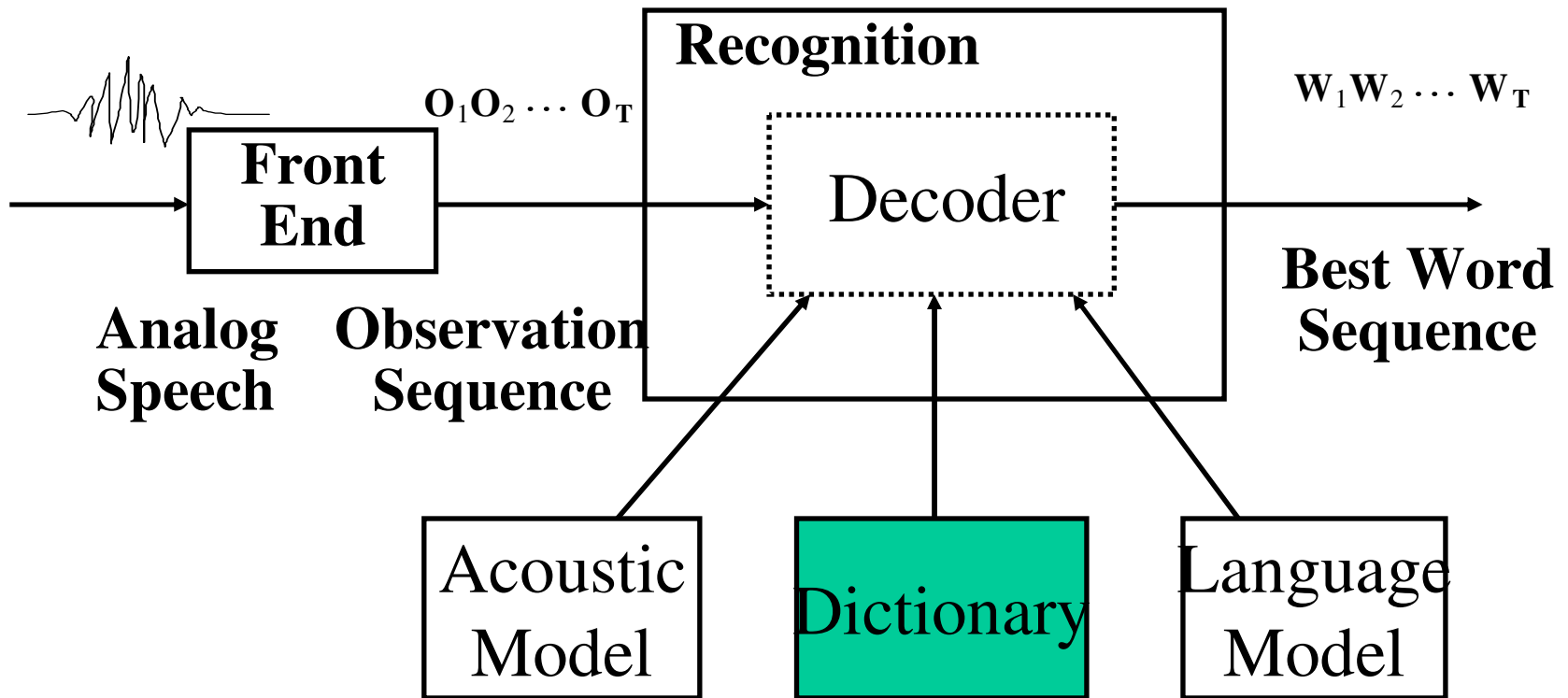
most common
approach:
Multi-Layer
Perceptron
MLP



- we can prove that a MLP can approximate the probabilities $P(\text{Class}|\text{Pattern})$
- most common training procedure: **error backpropagation**

Speech Recognition (System Components)

- Recognizer Components:



Dictionaries

- Word Dictionaries
 - Words and Word Models
 - Assign Certain Number of States to Each Word Model
- Phonetic Dictionaries
 - Convert Orthography to Phoneme Strings
 - Represent Alternate Pronunciations
 - “Zero”, “Oh”, “Because”, “Cause”
 - Multiwords: “Did You” → “Didjah”
- Tree-Structured Dictionary
 - Faster Search, Faster Overall Run-Time