# Hidden Markov Model and Speech Recognition

Nirav S. Uchat

1 Dec,2006

# Outline

# Introduction

## What is Speech Recognition ?

- Understanding what is being said
- Mapping speech data to textual information

## Speech Recognition is indeed challenging

- Due to presence of noise in input data
- Variation in voice data due to speaker's physical condition, mood etc..
- Difficult to identify boundary condition

# Different types of Speech Recognition

- Type of Speaker
  - Speaker Dependent(SD)
    - relatively easy to construct
    - requires less training data (only from particular speaker)
    - also known as speaker recognition
  - Speaker Independent(SID)
    - requires huge training data (from various speaker)
    - difficult to construct
- Type of Data
  - Isolates Word Recognizer
    - recognize single word
    - easy to construct (pointer for more difficult speech recognition)
    - may be speaker dependent or speaker independent
  - Continuous Speech Recognition
    - most difficult of all
    - problem of finding word boundary

# Outline

# Use of Signal Model

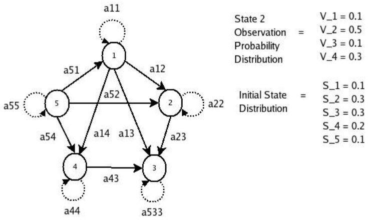
- it helps us to characterize the property of the given signal
- provide theoretical basis for signal processing system
- way to understand how system works
- we can simulate the source and it help us to understand as much as possible about signal source

## Why Hidden Markov Model (HMM) ?

- very rich in mathematical structure
- when applied properly, work very well in practical application

# Outline

## Components of HMM [2]




1. Number of state = N
2. Number of distinct observation symbol per state = M, $V = V_1, V_2, \cdots, V_M$
3. State transition probability $= a_{ij}$
4. Observation symbol probability distribution in state j, $B_j(K) = P[V_k \ at \ t | q_t = S_j]$
5. The initial state distribution $\pi_i = P[q_1 = S_i] \ \ 1 \leq i \leq N$

# Problem For HMM : Problem 1 [2]

- **Problem 1 : Evaluation Problem** Given the observation sequence $O = O_1\ O_2\ \cdots\ O_T$, and model $\lambda = (A, B, \pi)$, how do we efficiently compute $P(O|\lambda)$, the probability of observation sequence given the mode.

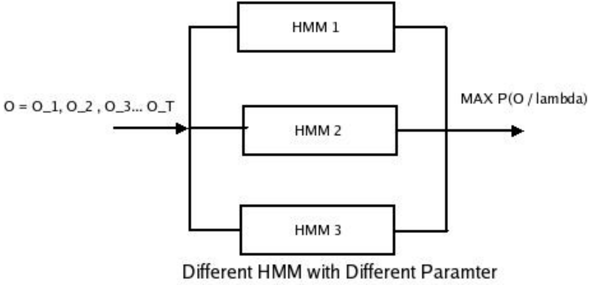



Figure: Evaluation Problem

# Problem 2 and 3 [2]

- **Problem 2 : Hidden State Determination (Decoding)** Given the observation sequence $O = O_1\ O_2\ \cdots\ O_T$, and model $\lambda = (A, B, \pi)$, How do we choose "BEST" state sequence $Q = q_1\ q_2\ \cdots\ q_T$ which is optimal in some meaningful sense.
  (In Speech Recognition it can be considered as state emitting correct phoneme)
- **Problem 3 : Learning** How do we adjust the model parameter $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$. Problem 3 is one in which we try to optimize model parameter so as to best describe as to how given observation sequence comes out

# Solution For Problem 1 : Forward Algorithm

- $P(O|\lambda) = \sum_{q_1, \cdots, q_T} \pi_{q_1} b_{q1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \cdots a_{q_{T-1} q_T} b_{q_T}(O_T)$
- Which is O($N^T$) algorithm i.e. at every state we have N choices to make and total length is T.
- Forward algorithm which uses dynamic programming method to reduce time complexity.
- It uses forward variable $\alpha_t(i)$ defined as

  $$\alpha_t(i) = P(O_1, O_2, \cdots, O_i, q_t = S_i|\lambda)$$

  i.e., the probability of partial observation sequence, $O_1, O_2$ till $O_t$ and state $S_i$ at time $t$ given the model $\lambda$,

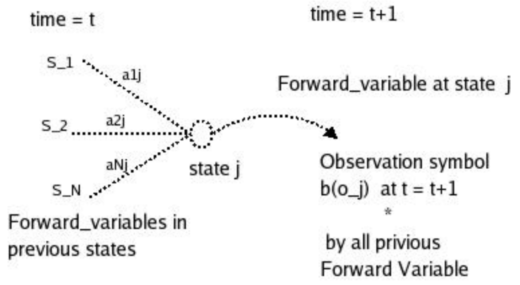



Figure: Forward Variable

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(O_{t+1}), \qquad 1 \leq t \leq T-1, \quad 1 \leq j \leq N$$

# Solution For Problem 2 : Decoding using Viterbi Algorithm [1]

- Viterbi Algorithm : To find single best state sequence
- we define a quantity

$$\delta_t(i) = \max_{q_1, q_2, \cdots, q_{t-1}} P[q_1 q_2 \cdots q_t = i, O_1 O_2 \cdots O_t | \lambda]$$

  i.e., $\delta_t(i)$ is the best score along a single path, at time $t$, which account for the first $t$ observations and ends in state $S_i$, by induction

$$\delta_{t+1}(j) = \left[ \max_i \delta_t(i) a_{ij} \right] b_j(O_{t+1})$$

- Key point is, **Viterbi algorithm** is similar (except for the backtracking part) in implementation to the **Forward algorithm**. The major difference is maximization of the previous state in place of summing procedure in forward calculation

# Solution For Problem 3 : Learning (Adjusting model parameter)

- Uses Baum-Welch Learning Algorithm
- Core operation is
  - $\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda)$ i.e.,the probability of being in state $S_i$ at time $t$, and state $S_j$ at time $t+1$ given the model and observation sequence
  - $\gamma_t(i) =$ the probability of being in state $S_i$ at time $t$, given the observation sequence and model
  - we can relate :

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j)$$

  - re-estimated parameters are :

$$\bar{\pi} = \text{Expected number of times in state } S_i = \gamma_1(i)$$

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

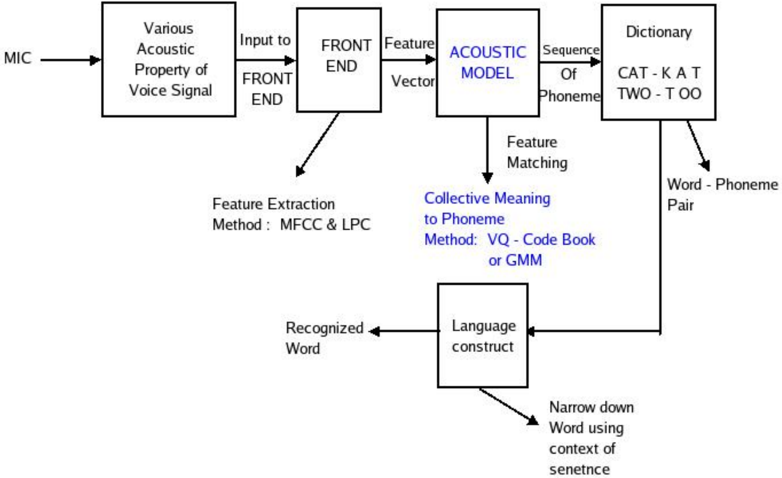
$$= \frac{\displaystyle\sum_{t=1}^{T-1} \xi_t(i,j)}{\displaystyle\sum_{t=1}^{T-1} \gamma_t(i)}$$

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

$$= \frac{\displaystyle\sum_{\substack{t=1 \\ s.t\ O_t = V_k}}^{T} \gamma_t(j)}{\displaystyle\sum_{t=1}^{T} \gamma_t(j)}$$

# Outline

# Block Diagram of ASR using HMM

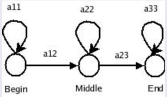


# Basic Structure

## Phoneme

- smallest unit of information in speech signal (over 10 msec) is Phoneme
- “ONE” : W AH N
- English language has approximately 56 phoneme

## HMM structure for a Phoneme




- This model is First Order Markov Model
- Transition is from previous state to next state (no jumping)

# Question to be ask ?

## What represent state in HMM ?

- HMM for each phoneme
- 3 state for each HMM
- states are : start mid and end
- "ONE" : has 3 HMM for phoneme W AH and N each HMM has 3 state

## What is output symbol ?

- Symbol form Vector Quantization is used as output symbol from state
- concatenation of symbol gives phoneme

# Front-End

purpose is to parameterize an input signal (e.g., audio) into a sequence of Features vector

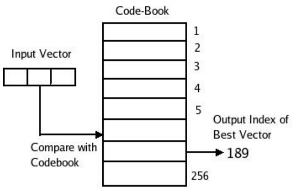
Method for Feature Vector extraction are

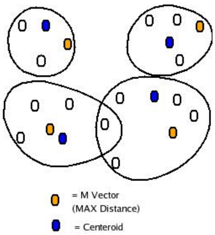
- MFCC - Mel Frequency Cepsteral Coefficient
- LPC Analysis - Linear Predictive Coding

# Acoustic Modeling[1]

Uses Vector Quantization to map Feature vector to Symbol.

- create training set of feature vector
- cluster them in to small number of classes
- represent each class by symbol
- for each class $V_k$, compute the probability that it is generated by given HMM state.

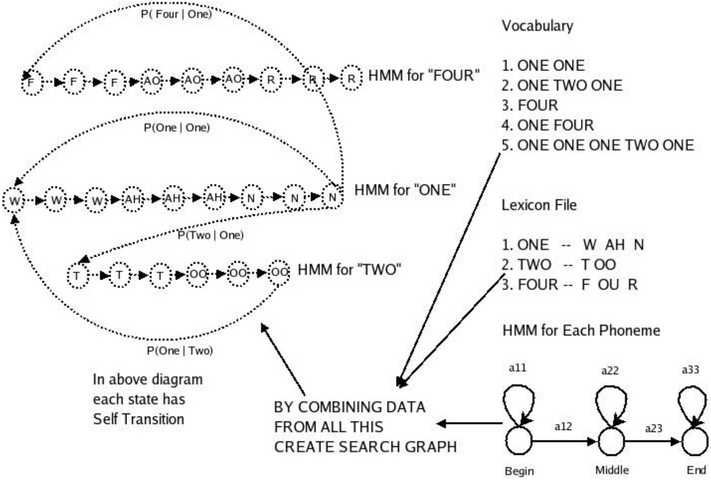


# Creation Of Search Graph [3]

- Search Graph represent Vocabulary under consideration
- Acoustic Model, Language model and Lexicon (Decoder during recognition) works together to produce Search Graph
- Language model represent how word are related to each other (which word follows other)
- it uses Bi-Gram model
- Lexicon is a file containing WORD – PHONEME pair
- So we have whole vocabulary represented as graph

# Complete Example



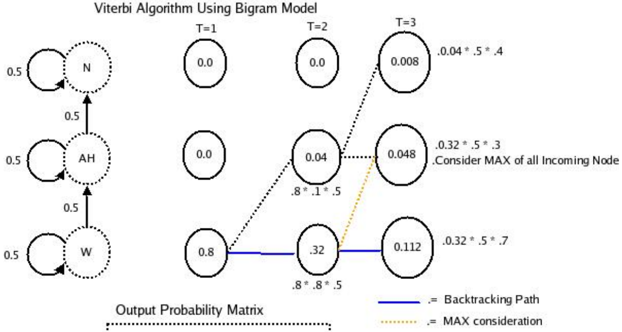
# Training

Training is used to adjust model parameter to maximize the probability of recognition

- Audio data from various different source are taken
- it is given to the prototype HMM
- HMM will adjust the parameter using Baum-Welch algorithm
- Once the model is train, unknown data is given for recognition

# Decoding

It uses Viterbi algorithm for finding “BEST” state sequence




Output Probability Matrix

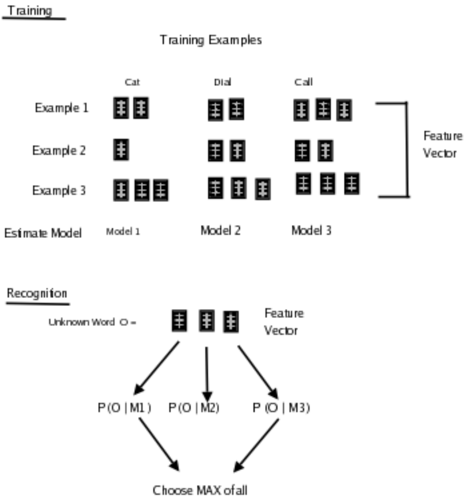
| Time | 1 | 2 | 3 |
|---|---|---|---|
| W | 0.8 | 0.8 | 0.7 |
| AH | 0.1 | 0.1 | 0.3 |
| N | 0.6 | 0.6 | 0.4 |

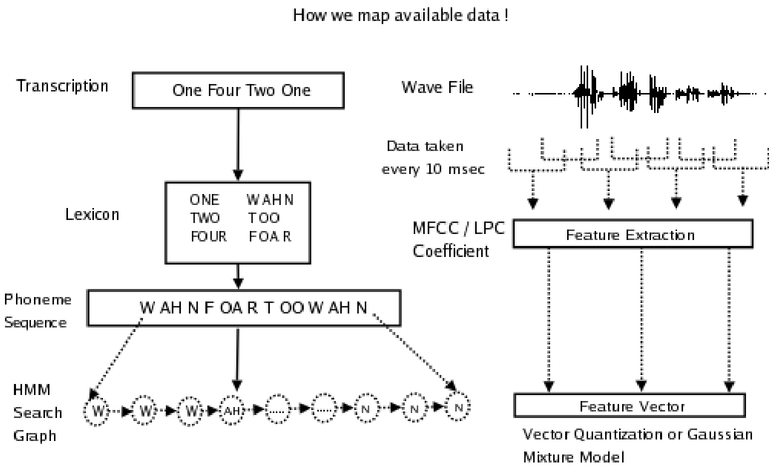
# Decoding Continued

- This is just for Single Word
- During Decoding whole graph is searched.
- Each HMM has two non emitting state for connecting it to other HMM

# Outline

# Isolated Word Recognizer [4]




How we map available data !
Transcription
One Four Two One
Lexicon
ONE WAHN
TWO TOO
FOUR FOAR
Phoneme Sequence
W AH N F OA R T OO W AH N
HMM Search Graph
W
W
W
AH
.....
.....
N
N
N
Wave File
Data taken every 10 msec
MFCC / LPC Coefficient
Feature Extraction
Feature Vector
Vector Quantization or Gaussian Mixture Model

# Problem With Continuous Speech Recognition

- Boundary condition
- Large vocabulary
- Training time
- Efficient Search Graph creation

- Dan Jurafsky.
  *CS 224S / LINGUIST 181 Speech Recognition and Synthesis.*
  World Wide Web, http://www.stanford.edu/class/cs224s/.
- Lawrence R. Rabiner.
  *A Tutorial on Hidden Markov Model and Selected Applicaiton in Speech Recognition.*
  IEEE, 1989.
- Willie Walker, Paul Lamere, and Philip Kwok.
  *Sphinx-4: A Flexible Open Source Framework for Speech Recognition.*
  SUN Microsystem, 2004.
- Steve Young and Gunnar Evermannl.
  *The HTK Book.*
  Microsoft Corporation, 2005.