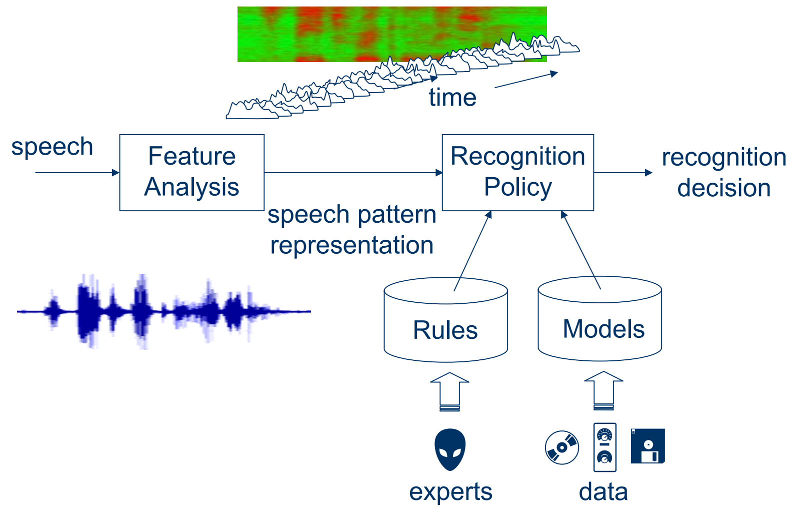
Hidden Markov Model and Its Applications in Speech Recognition – A Tutorial

B.H. Juang
Georgia Institute of Technology

Speech Recognition Techniques



Statistical Pattern Recognition Problem

Problem Statement:
To recognize/classify an unknown observation X as one of M classes (of events or species) with minimum ***probability of error***

Definition of error and error probability -
Conditional Error: given X, the risk associated with deciding that it is a class i event

$$R(C_i \mid X) = \sum_{j=1}^{M} e_{ij} P(C_j \mid X)$$

$P(C_j \mid X) =$ probability that X (the given observation) is a class j event
e_{ij} is the cost of classifying a class j event as a class i event; usually $e_{ij} \geq 0, e_{ii} = 0$

Expected Error:

$$\mathcal{E} = \int R(C(X) \mid X) p(X) dX$$

where $C(X)$ is the decision (based on a policy) made on X

How should $C(X)$ be made to achieve minimum error probability?

Bayes Decision Theory

Note: $R(C_i \mid X) = \sum_{j=1}^{M} e_{ij} P(C_j \mid X)$ and suppose $e_{ij} = 1, e_{ii} = 0$

If we institute the policy: $C_{MAP}(X) = C_i = \arg\max_{C_j} P(C_j \mid X)$

then $R(C_{MAP}(X) \mid X) = \min_{C_j} R(C_j \mid X)$

Why? $R(C_m \mid X) = \sum_{j=1, j\neq m}^{M} e_{ij} P(C_j \mid X) > R(C_n \mid X) = \sum_{j=1, j\neq n}^{M} e_{ij} P(C_j \mid X)$

if $P(C_m \mid X) < P(C_n \mid X)$

So, the best policy is $C(X) = C_i = \arg\max_{C_j} P(C_j \mid X)$

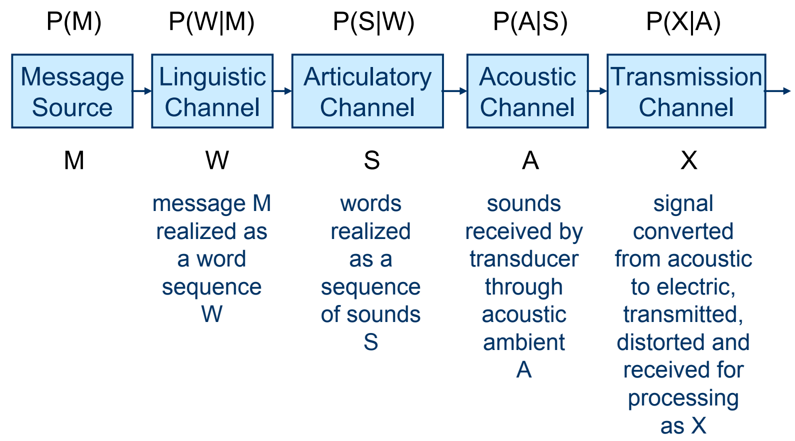
It is the so-called Maximum A Posteriori (MAP) decision.

Caveat: How do we know all $P(C_i \mid X),\ i = 1, 2, \cdots, M$ for any X?

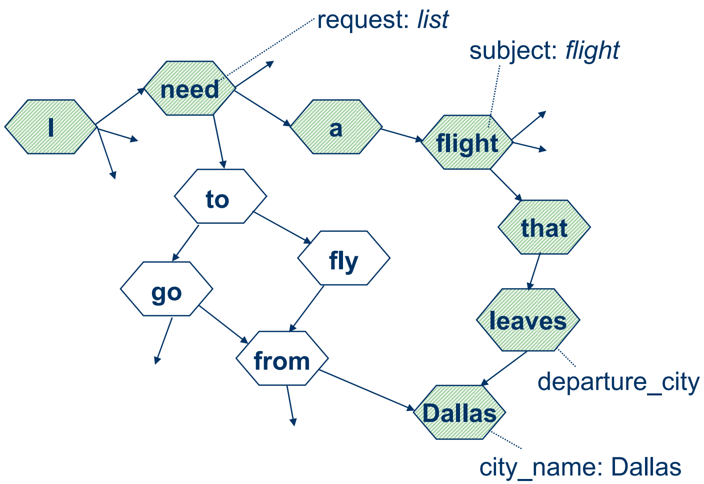
Representation of Speech & Speech Signal

- Grammar & Syntax
  - How the occurrence of words in sequence is governed
- Lexicon/Dictionary
  - How a word is supposed to be pronounced as a sequence of unitary sounds
- Acoustic-phonetics
  - How a unitary sound and/or a sequence of unitary sounds are supposed to be produced with the articulatory apparatus

Models for Production of Speech

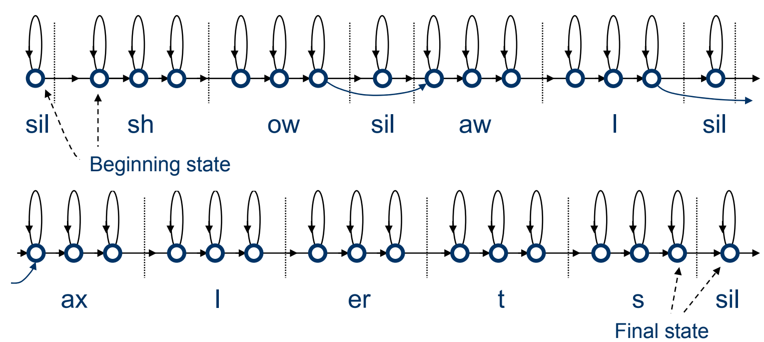


Language as a Finite State Machine



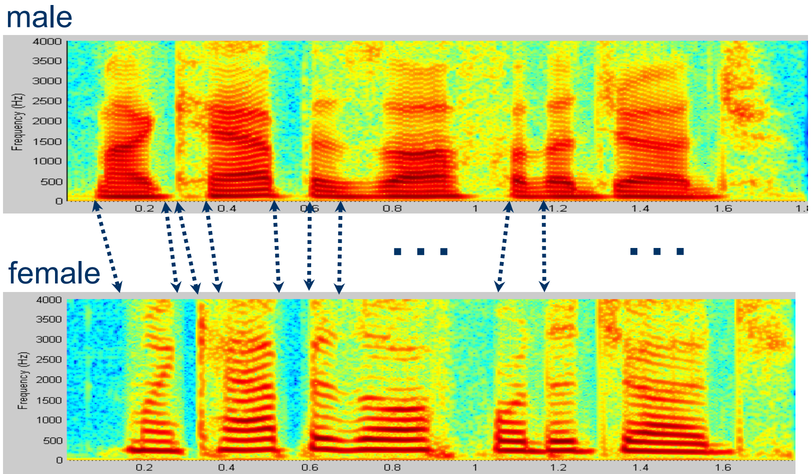
Lexicon & Phonology also as an FSM or FSN

COMPOSITE Finite State Network

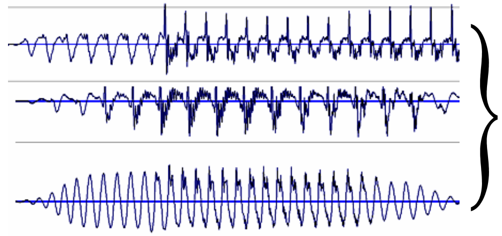


context-dependent phoneme model examples:
ϕ-**sh**-ow, ϕ-**ax**-l, ax-**l**-er, l-**er**-t

Temporal Variation in Speech



Variations in Speech

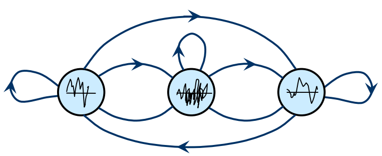


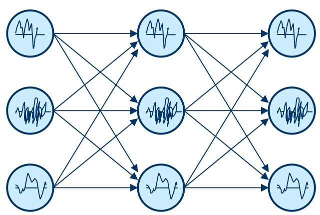
These are three real waveforms for the word "**`my`**"; they are very different.

How to put these vastly different realizations in the same stochastic model so as to allow meaningful identification of the process (as word "**`my`**")?

A doubly stochastic process called
mixture distribution Hidden Markov Model

Hidden Markov Model





Speech observation

$$\mathbf{x} = (x_1, x_2, \cdots, x_T)$$

State sequence

$$\mathbf{q} = (q_0, q_1, q_2, \cdots, q_T)$$

$$p(\mathbf{x} \mid c; \Lambda) = \sum_{\mathbf{q}} p(\mathbf{x}, \mathbf{q} \mid c; \Lambda)$$

$$p(\mathbf{x}, \mathbf{q} \mid c; \Lambda) = \pi_0 \prod_{t=1}^{T} a_{q_{t-1} q_t} b_{q_t}(x_t)$$

- Each state represents a process of measurable observations;
- Inter-process transition is governed by a finite state Markov chain;
- Processes are stochastic and individual observations do not immediately identify the state.

Hidden Markov Models - Specifications

$\mathbf{x} = (x_1, x_2, \cdots, x_T)$ is the sequence of observations
$\mathbf{q} = (q_0, q_1, \cdots, q_T)$ is the sequence of states the system is in

- Number of states of the Markov chain, N
- State transition probability matrix, $A = \left[a_{ij} \right]_{N \times N}$
  $a_{ij} = \Pr[q_t = j \mid q_{t-1} = i]$ $\quad \sum_{j=1}^{N} a_{ij} = 1$ for all i
- In-state observation probability distribution functions
  $B = \{b_i(x)\}_{i=1}^{N}$ $\quad b_i(x) \Rightarrow b_i(x, \lambda_i)$ i.e., parameterized by λ_i
- Initial state probability distribution,
  $\boldsymbol{\pi}^t = [\pi_1, \pi_2, \cdots, \pi_N]$ where $\pi_i = \Pr[q_0 = i]$ $\quad \sum_{i=1}^{N} \pi_i = 1$

The triple $\Lambda = (\boldsymbol{\pi}, A, B)$ defines a hidden Markov model.

Three Basic Problems of HMM

- Given the observation sequence $\mathbf{x} = (x_1, x_2, \cdots, x_T)$ and a model $\Lambda = (\boldsymbol{\pi}, A, B)$, how do we efficiently compute $P(\mathbf{x}; \Lambda)$?
- Given the observation sequence $\mathbf{x} = (x_1, x_2, \cdots, x_T)$ and the model $\Lambda = (\boldsymbol{\pi}, A, B)$, how do we find a corresponding state sequence $\mathbf{q} = (q_0, q_1, q_2, \cdots, q_T)$ that is optimal in some sense?
- Given an observation sequence X, or a number of sequences $\{\mathbf{x}^{(i)}\}_i$, how to estimate parameters in the model set $\Lambda = (\boldsymbol{\pi}, A, B)$?

Evaluation of HMM Probability

$$P(\mathbf{x}; \Lambda) = \sum_{\mathbf{q}} P(\mathbf{x}, \mathbf{q}; \Lambda) \qquad P(\mathbf{x}, \mathbf{q}; \Lambda) = \pi_{q_0} \prod_{t=1}^{T} a_{q_{t-1} q_t} b_{q_t}(x_t)$$

$$P(\mathbf{x}; \Lambda) = \sum_{\mathbf{q}} P(\mathbf{x}, \mathbf{q}; \Lambda) = \sum_{\mathbf{q}} \pi_{q_0} \prod_{t=1}^{T} a_{q_{t-1} q_t} b_{q_t}(x_t)$$

Direct evaluation will involve $2T \bullet N^T$ calculations.

The Forward Procedure

Define $\alpha_t(i) = \Pr\{(x_1, x_2, \cdots, x_t, q_t = i; \Lambda)$

$\alpha_t(i)$ Is the probability of the partial observation sequence $x_1, x_2, \cdots, x_t$, up to time t , and the system is at state i at time t .

Forward Procedure

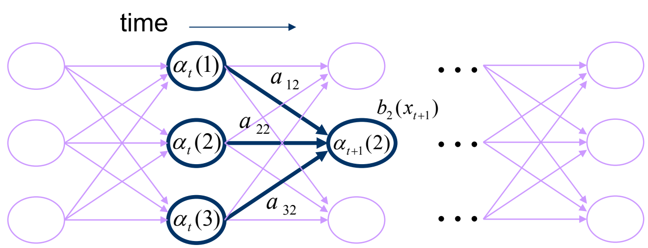
Initialization: $\alpha_0(i) = \pi_i, \quad i = 1, 2, \cdots, N$

Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(x_{t+1}), \quad 0 \le t \le T-1, 1 \le j \le N$$

Termination: $P(\mathbf{x}; \Lambda) = \sum_{i=1}^{N} \alpha_T(i)$

~$N^2 T$ calculations
Linear in T



Optimal State Sequence

Several possibilities

- The state sequence that maximizes the joint state-observation probability

$$\mathbf{q}_{opt} = \arg\max_{\mathbf{q}} P(\mathbf{x}, \mathbf{q}; \Lambda) = \arg\max_{\mathbf{q}} \pi_{q_0} \prod_{t=1}^{T} a_{q_{t-1} q_t} b_{q_t}(x_t)$$

- The state sequence that consists of individual states maximizing the a posteriori probability given the observation

$\gamma_t(i) = P(q_t = i \mid \mathbf{x}; \Lambda)$ i.e. probability of being in state i at time t, given $\mathbf{x}$

$$\gamma_t(i) = P(q_t = i \mid \mathbf{x}; \Lambda) = P(\mathbf{x}, q_t = i; \Lambda)[P(\mathbf{x}; \Lambda)]^{-1}$$
$$= P(\mathbf{x}, q_t = i; \Lambda)\left[\sum_{i=1}^{N} P(\mathbf{x}, q_t = i; \Lambda)\right]^{-1}$$

Parameter Estimation

Maximum Likelihood Estimation

➔ find Λ to maximize $p(\mathbf{x} \mid c; \Lambda)$

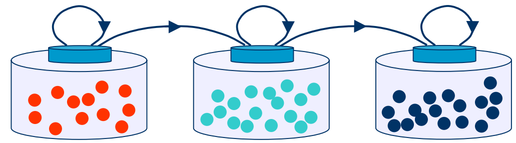
Estimation Algorithms:

- $b_i(x)$ is discrete - Baum & Egan, 1967
- $b_i(x)$ is log-concave continuous - Baum, Petrie, Soules and Weiss, 1970
- $b_i(x)$ is elliptically symmetric - Liporace, 1982
- $b_i(x)$ is mixture of log-concave or elliptically symmetric – Bell Labs, 1984

Continuous mixture density hidden Markov model *can approximate any density function with arbitrary precision, provided that the number of mixture components is unconstrained.*

From Markov Chain to Discrete HMM

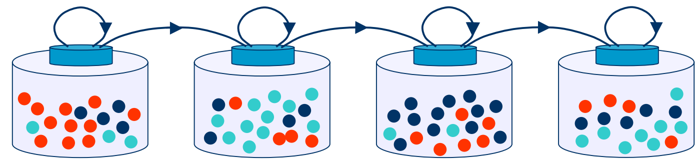
A 3-state Markov Chain



$b_1(x) = \delta(x = Rd)$ $b_2(x) = \delta(x = Bl)$ $b_3(x) = \delta(x = Bk)$

A 4-state discrete hidden Markov model

N = 4
M = 3



$b_1(x) = P_1(Rd)\delta(x-Rd) + P_1(Bl)\delta(x-Bl) + P_1(Bk)\delta(x-Bk)$

$b_2(x) = P_2(Rd)\delta(x-Rd) + P_2(Bl)\delta(x-Bl) + P_2(Bk)\delta(x-Bk)$

$b_3(x) = P_3(Rd)\delta(x-Rd) + P_3(Bl)\delta(x-Bl) + P_3(Bk)\delta(x-Bk)$

$b_4(x) = P_4(Rd)\delta(x-Rd) + P_4(Bl)\delta(x-Bl) + P_4(Bk)\delta(x-Bk)$

In-State (Local) Observation Distributions

- Discrete distributions

  $x \in \{s_k\}_{k=1}^{M}$ and $b_i(x = s_k) = \Pr\{x = s_k, q = i\} = b_{ik}$

- Log-concave probability density functions

  x is continuous-valued and $\log b_i(x) = \log f_i(x)$ is a concave function

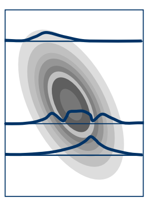
- Elliptically symmetric probability density functions

  $b_i(x) = \int f(x, g)\, d\mu(g)$

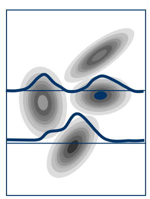
- General mixture probability density functions

$$b_i(x) = \sum_{k=1}^{M} c_{ik} f_{ik}(x)$$

where $\sum_{k=1}^{M} c_{ik} = 1$



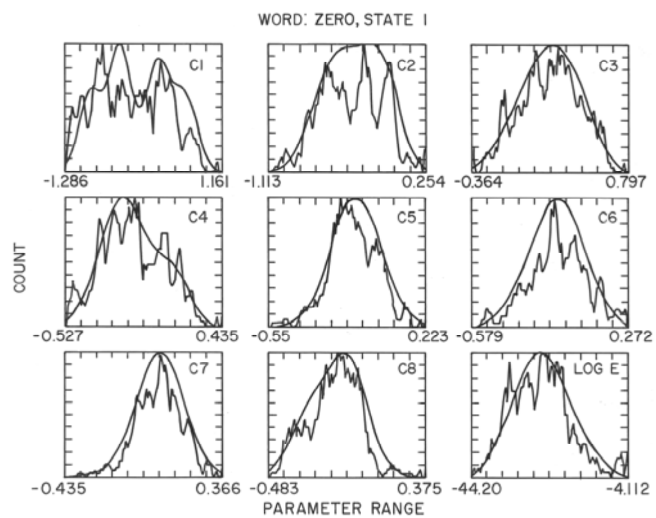
Elliptically Symmetric Distribution



Mixture of Elliptically Symmetric Distribution

marginal

Multivariate Mixture Distribution



Autoregressive HMM

Consider an observation vector (e.g., a frame of speech signal) $x=(x_0,x_1,x_2,\cdots,x_{K-1})$ where each x_k is a waveform sample.

We assume the source that produces x is an autoregressive one with the following governing equation

$$x_k=-\sum_{i=1}^{p} a_i x_{k-i}+e_k \quad 0\le k\le K-1$$

Recall LPC

Where e_k are Gaussian, independent, identically distributed random variables with zero mean and variance σ^2, and $\{a_i\}_{i=1}^p$ are the autoregressive or predictor coefficients.

As $K\,(\text{length of data})\to\infty$, then the pdf of x becomes

$$f(x)=(2\pi\sigma^2)^{-K/2}e^{-\delta(x,\mathbf{a})/(2\sigma^2)}$$

In short-time analysis context, each vector would carry a time index.

where $\delta(x,\mathbf{a})=r_a(0)r(0)+2\sum_{i=1}^{p} r_a(i)r(i)$

$$r_a(i)=\sum_{n=0}^{p-i} a_n a_{n+i},\quad a_0=1 \quad \text{and} \quad r(i)=\sum_{n=0}^{K-i-1} x_n x_{n+i},\quad 0\le i\le p$$

Mixture Autoregressive HMM

Each state is associated with M mixture components; each mixture component is defined by an autoregressive pdf:

$$b_j(x)=\sum_{m=1}^{M} c_{jm} b_{jm}(x) \quad \text{where} \quad b_{jm}(x)=(2\pi\sigma_{jm}^2)^{-K/2}e^{-\delta(x,\mathbf{a}_{jm})/(2\sigma_{jm}^2)}$$

Each distribution is characterized by an autocorrelation vector which in turn defines the predictor vector $\mathbf{a}_{jm}$. In re-estimation, the transformation on autocorrelation vector (for each mixture component) is to obtain an average of the autocorrelation vectors, each weighted by the corresponding probability of being associated with the particular mixture component

$$\bar{\mathbf{r}}_{jm}=\frac{\sum_{t=1}^{T}\gamma_t(j,m)\mathbf{r}_t}{\sum_{t=1}^{T}\gamma_t(j,m)} \qquad \gamma(j,m)=\left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)}\right]\left[\frac{c_{jm}b_{jm}(x_t)}{\sum_{m=1}^{M}c_{jm}b_{jm}(x_t)}\right]$$

ML Re-estimation – a.k.a. EM

Define $Q(\Lambda,\Lambda')=\sum_{\mathbf{q}} P(\mathbf{x},\mathbf{q};\Lambda)\log P(\mathbf{x},\mathbf{q};\Lambda')$ $\quad$ Q = Auxiliary function

Theorem: If $Q(\Lambda,\Lambda')\ge Q(\Lambda,\Lambda)$ then $P(\mathbf{x};\Lambda')\ge P(\mathbf{x};\Lambda)$. The inequality is strict unless $P(\mathbf{x},\mathbf{q};\Lambda')\ge P(\mathbf{x},\mathbf{q};\Lambda)$ almost everywhere.

Re-estimation:

1. Given Λ, define the auxiliary function as a function of Λ';
2. Maximize the auxiliary function over Λ' and obtain $\bar{\Lambda}$

$$\bar{\Lambda}=T(\Lambda)\in\Psi=\{\hat{\Lambda}\,|\,Q(\Lambda,\hat{\Lambda})=\max_{\Lambda'}Q(\Lambda,\Lambda')\}$$

3. Replace Λ with $\bar{\Lambda}$ and repeat the above until a stationary point is reached.

This is the Baum-Welch re-estimation, a hill-climbing algorithm to achieve ML, similar to the EM (expectation-maximization) algorithm.

Reestimation Transformation

$$\bar{\Lambda}=T(\Lambda)\in\Psi=\{\hat{\Lambda}\,|\,Q(\Lambda,\hat{\Lambda})=\max_{\Lambda'}Q(\Lambda,\Lambda')\}$$

For Gaussian mixture density HMM: $\quad b_i(x)=\sum_{k=1}^{M}c_{ik}f(x;\mu_{ik},\Sigma_{ik})$

Initial state probability: $\bar{\pi}_i=P(\mathbf{x},q_0=i;\Lambda)[P(\mathbf{x};\Lambda)]^{-1}$

State transition probability:

$$\bar{a}_{ij}=\sum_{t=1}^{T}P(\mathbf{x},q_{t-1}=i,q_t=j;\Lambda)\left[\sum_{t=1}^{T}P(\mathbf{x},q_{t-1}=i;\Lambda)\right]^{-1}$$

Mixture weights:

$$\bar{c}_{ik}=\sum_{t=1}^{T}P(\mathbf{x},q_t=i,u_t=k;\Lambda)\left[\sum_{t=1}^{T}P(\mathbf{x},q_t=i;\Lambda)\right]^{-1}$$

Gaussian parameters:

$$\bar{\mu}_{ik}=\sum_{t=1}^{T}x_t P(\mathbf{x},q_t=i,u_t=k;\Lambda)\left[\sum_{t=1}^{T}P(\mathbf{x},q_t=i,u_t=k;\Lambda)\right]^{-1}$$

$$\bar{\Sigma}_{ik}=\sum_{t=1}^{T}(x_t-\mu_{ik})(x_t-\mu_{ik})^t P(\mathbf{x},q_t=i,u_t=k;\Lambda)\left[\sum_{t=1}^{T}P(\mathbf{x},q_t=i,u_t=k;\Lambda)\right]^{-1}$$

Interpretation of Re-estimation Formula

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T} P(\mathbf{x}, q_{t-1} = i, q_t = j; \Lambda = \{\pi, A = [a_{ij}], B\})}{\sum_{t=1}^{T} P(\mathbf{x}, q_{t-1} = i; \Lambda = \{\pi, A = [a_{ij}], B\})}$$

Probability version of $\dfrac{\text{count(transition from state } i \text{ to state } j)}{\text{count(transition from state } i)}$

$$\bar{c}_{ik} = \frac{\sum_{t=1}^{T} P(\mathbf{x}, q_t = i, u_t = k; \Lambda)}{\sum_{t=1}^{T} P(\mathbf{x}, q_t = i; \Lambda)}$$

Probability version of $\dfrac{\text{count(transition from state } i \text{ along mixture } k)}{\text{count(transition from state } i)}$

$$\bar{\mu}_{ik} = \frac{\sum_{t=1}^{T} x_t P(\mathbf{x}, q_t = i, u_t = k; \Lambda)}{\sum_{t=1}^{T} P(\mathbf{x}, q_t = i, u_t = k; \Lambda)}$$

Expected (or probability-weighted) average or covariance along each mixture component in each state pdf.

$$\bar{\Sigma}_{ik} = \sum_{t=1}^{T} (x_t - \mu_{ik})(x_t - \mu_{ik})^t P(\mathbf{x}, q_t = i, u_t = k; \Lambda) \left[\sum_{t=1}^{T} P(\mathbf{x}, q_t = i, u_t = k; \Lambda) \right]^{-1}$$

Segmental K-Means Algorithm

Motivation:

derive good estimates of the $b_i(x)$ densities as required for rapid convergence of re-estimation procedure.

Initially:

training set of multiple sequences of observations, initial model estimate.

Procedure:

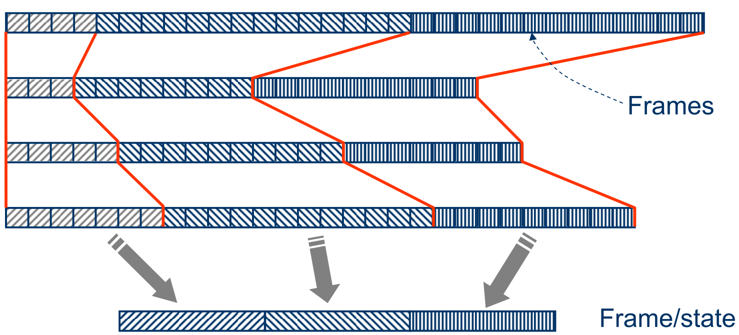
segment each observation sequence into states using a Viterbi procedure. For discrete observation densities, code all observations in state *j* using the *M*-codeword codebook, giving

$b_i(x)$ = number of vectors with codebook index *k*, in state *j*, divided by the number of vectors in state *j*.

for continuous observation densities, cluster the observations in state *j* into a set of *M* clusters, giving

$$\bar{\Lambda} = \arg\max_{\Lambda} \max_{\mathbf{q}} p(\mathbf{x}, \mathbf{q} \mid c; \Lambda)$$

Getting the Right Statistics



- Segmental representations;
- Take advantage of similarity between adjacent frames to derive stable representations
- Take advantage of many tokens to derive consistent representations

Beyond Maximum Likelihood HMM

Two Motivating Questions

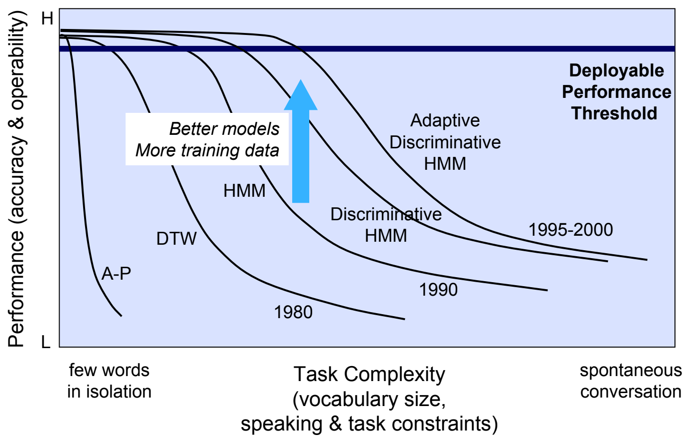
1. **Have we been able to choose the "right" models for speech recognition?**

   No, we can't even agree on the "right" speech representation (I.e. the observation space). We are still working on the front-end, the transformation, and many other related issues. ***Efficiency*** is also critical.

2. **If not, what is the alternative recognizer design principle to follow?**

   We need to reexamine the role and basis of distribution estimation in recognizer design, reaffirm the goal of speech recognition (being to have the least recognition errors), and re-formulate the problem and strategy so as to obtain the best performance at minimum cost and highest efficiency.

Performance Issue

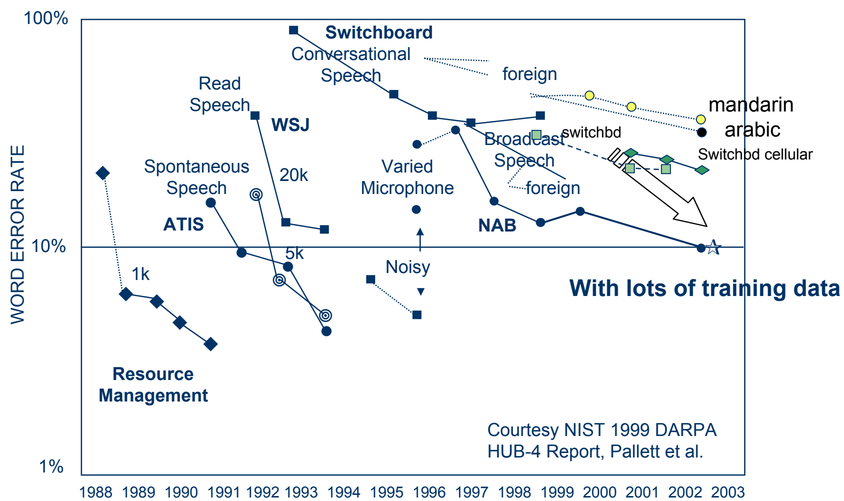


Typical Word Error Rates

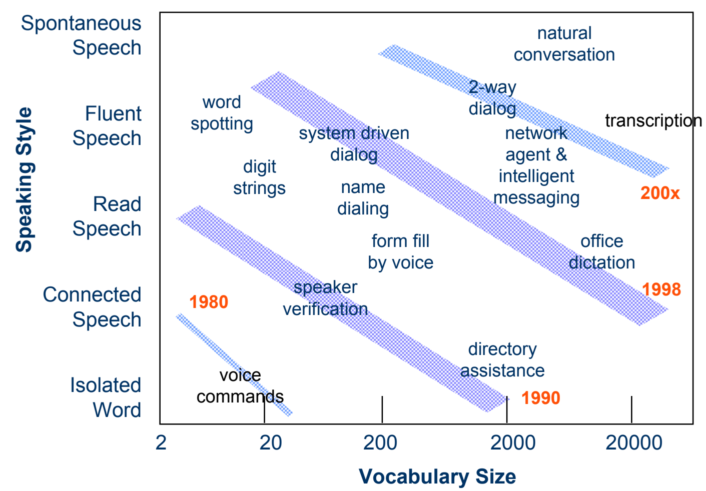
CORPUS	TYPE	VOCABULARY SIZE	WORD ERROR RATE
Connected Digit Strings--TI Database	Spontaneous	11 (zero-nine, oh)	0.3%
Connected Digit Strings--Mall Recordings	Spontaneous	11 (zero-nine, oh)	2.0%
Connected Digits Strings--HMIHY	Conversational	11 (zero-nine, oh)	5.0%
RM (Resource Management)	Read Speech	1000	2.0%
ATIS(Airline Travel Information System)	Spontaneous	2500	2.5%
NAB (North American Business)	Read Text	64,000	6.6%
Broadcast News	News Show	210,000	13-17%
Switchboard	Conversational Telephone	45,000	25-29%
Call Home	Conversational Telephone	28,000	40%

factor of 17 increase in digit error rate

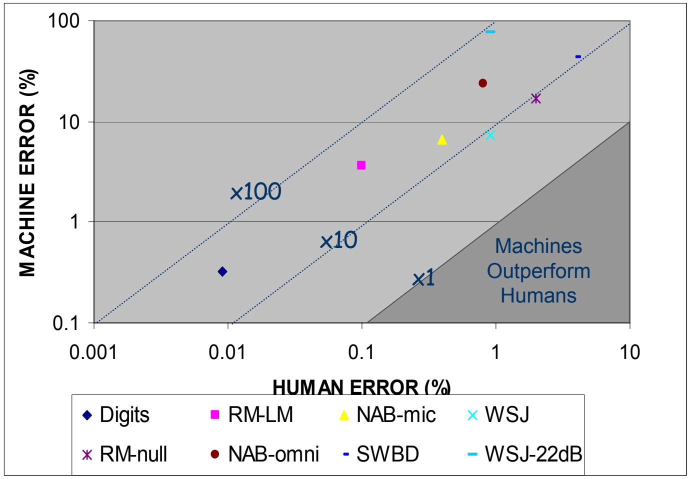
DARPA Speech Recognition Benchmark



Speech Recognition Progress



Human Speech Recognition vs ASR



ASR Challenges Ahead

- Variability of sounds (e.g., pronunciation, words, phrases)
  - within a single speaker
  - across speakers
  - across various microphones
  - transmission channels
- Background noise
  - road noise, fan, “constant” noise
  - background conversation; door slam
- Speaker production errors
  - hesitations, extraneous speech
- Speech related effects
  - minimally distinct words
  - word/sound co-articulation
- The language
- Natural expressions in speech conversation

Structure of model

Robustness & adaptation of model

Pronunciation Variations

In a typical Switchboard data set (based on a DARPA report by Dragon Systems)

Reference Dictionary - constructed from Call-home and Switchboard

3M words training set of 28,000 distinct words, 3500 of which have multiple pronunciations.

Test Data Set -

- 4700 word tokens; 900 distinct words
- 2100 pronunciations according to phonetic transcription
- 2200 tokens (47%) pronounced “properly” according to dictionary
- 1500 new pronunciations emerge for complete coverage
- Other attributes:
  - 650 words with single pronunciation
  - “*the*” has 36 pronunciations
  - *schwa* is pronunciation of 27 words; 38 pronunciations are homonymic with more than 5 words
  - “the” and “to” are most confusable with 7 pronunciations in common

(Limited) Spoken Language Understanding

- **I**nterpret the **meaning** of key words and phrases in the recognized speech string, and map them to actions that the speech understanding system should take
  - accurate understanding can often be achieved without correctly recognizing every word in many limited tasks
- **Methodology:** exploit task grammar (syntax) and task semantics to **restrict the range of meaning** associated with the recognized word string; exploit ‘salient’ words and phrases to map high information word sequences to appropriate meaning
- **Applications:** automation of complex operator-based tasks, e.g., customer care, catalog ordering, form filling systems, provisioning of new services, customer help lines, etc.
- **Challenges:** what goes beyond simple classifications systems but below full Natural Language voice dialogue systems

Summary

- HMM is the dominant method in automatic speech recognition;
- Continuous mixture density HMM is the prevalent model structure that achieves best results in accuracy;
- Applications of HMM have been broadened to keyword spotting, speech understanding, and machine translation;
- Non-speech related applications are emerging as well.