# Introduction to Automatic Speech Recognition

Samudravijaya K
TIFR, samudravijaya@gmail.com

"Automatic Speech Recognition using Sphinx and HTK"
A hands-on Workshop
18-FEB-2011
AU-KBC Research Centre, Chennai

PROGRESS THROUGH KNOWLEDGE

http://www.au-kbc.org/speech http://speech.tifr.res.in

# Topics

- Overview
- Speech signal processing for feature extraction
- Recognition by Template matching
  - ∗ Vowel recognition
  - ∗ Classification of temporal patterns: DTW
- ASR using stochastic models
  - ∗ Acoustic model: HMM
  - ∗ Language model: Backoff trigram model

# What is ASR?

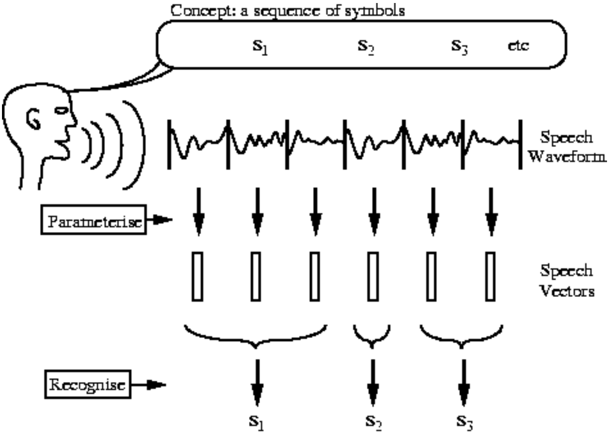


Fig. 1.1 Message Encoding/Decoding

source: HTK book

# Applications of ASR

Dictation machine

Command and Control

- Speech interface to computer
- Electronic gadgets: phone, TV, VCR etc.
- Eyes and hands busy situations: Car driver, Pilot in a cockpit
- Aids to handicapped: voice operated wheel chair
- Information retrieval: bank, travel, Telco
- Keyword spotting

# Types of ASR

Types of speech:

- Isolated Word Recognition (IWR)
- Connected Word Recognition (CWR)
- Continuous Speech Recognition (CSR)
- Spontaneous speech
- KeyWord Spotting (KWS)

Speaker dependence:

- speaker dependent/adaptive/independent
- multi-speaker

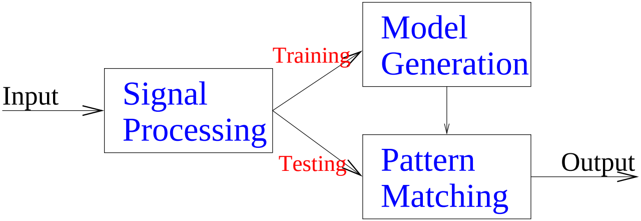
Vocabulary:

- Small ($<$ 100 words), Medium (hundreds), Large (thousands)
- Very large (tens of thousands), Out of vocabalary (OOV)

Bandwidth:

- Wideband/desktop
- Narrowband

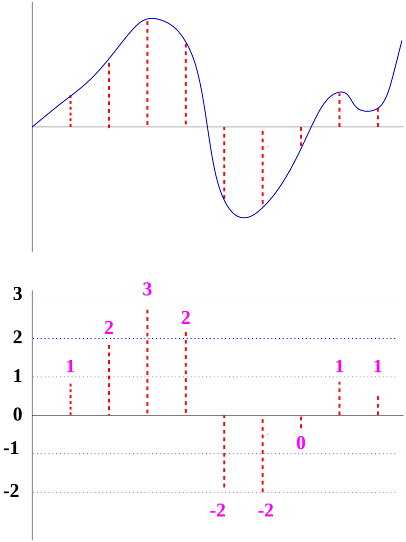
# Speech Recognition is Sequential Pattern Recognition



Goal: recognise the sequence of words from time waveform of speech.

Two phases: Training (learning) and Testing (recognition)

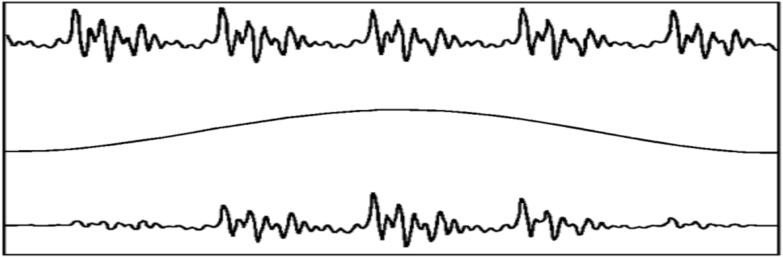
# Analog to digital (A2D) conversion



Digitisation = Sampling + Quantisation

analog wave −− > a sequence of integers

# Short-time processing

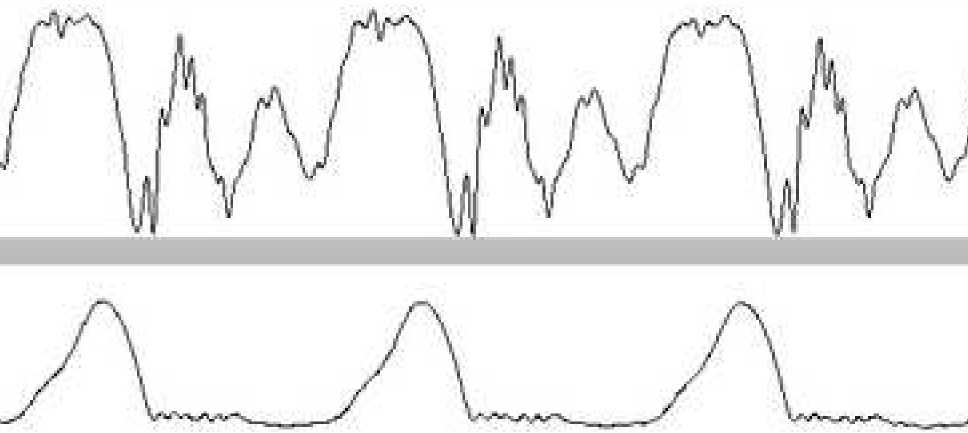


Blocking sequence into analysis **frames**

$$x(n) = s(m)w(n-m)$$

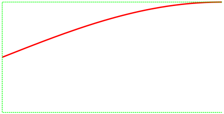
$w(n)$ is a *Tapering window*

# Production of voiced sounds



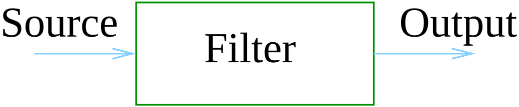
vowel अ

# Glottal impulses; Resonances of vocal tract



Uniform tube model

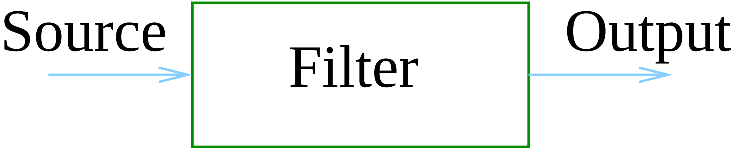
$$\nu = c/\lambda = 34000/4 * 17 = 500 Hz$$



glottal vibration vocal tract speech wave

**Formant === pole of a filter**

# Source-Filter model of speech production



Source

Filter

Output

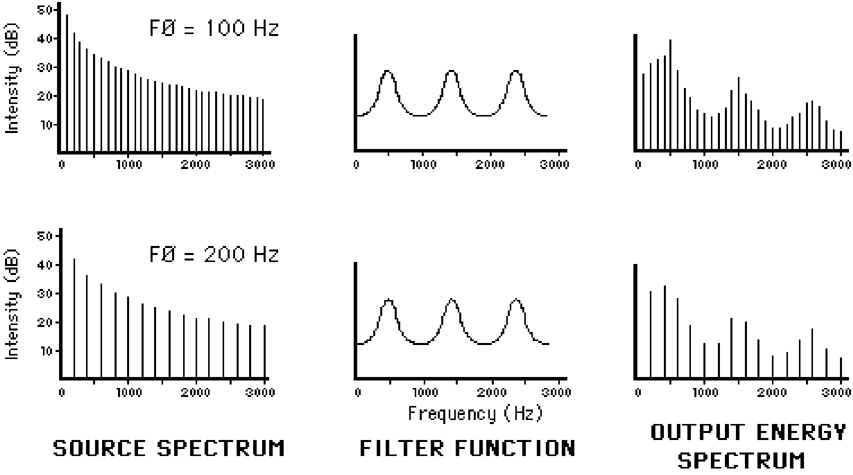
glottal vibration vocal tract speech wave

$$s(n) = e(n) * h(n)$$

$$S(k) = E(k)H(k)$$

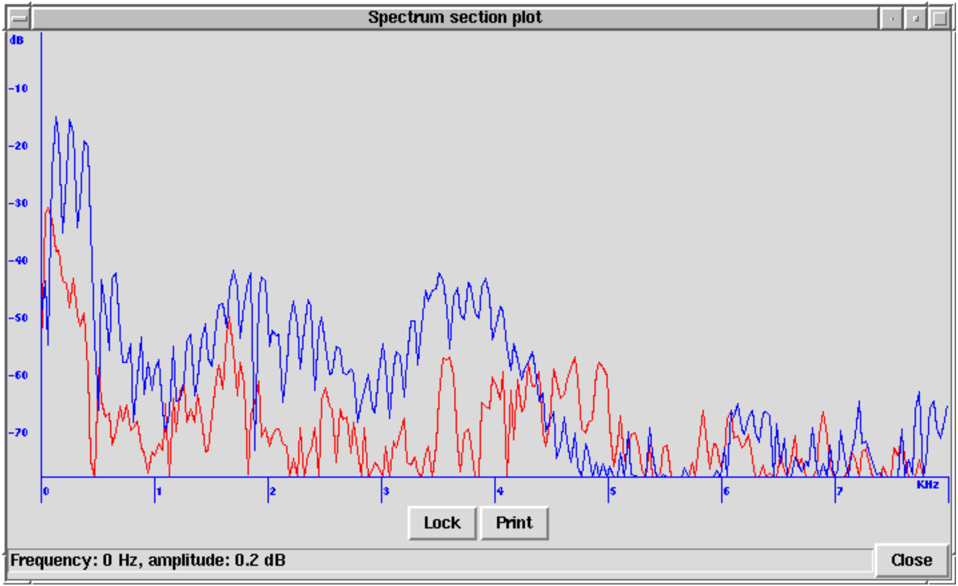
$$log(|S(k)| * *2) = log(|E(k)| * *2) \ + \ log(|H(k)| * *2)$$

# Illustration in spectral domain



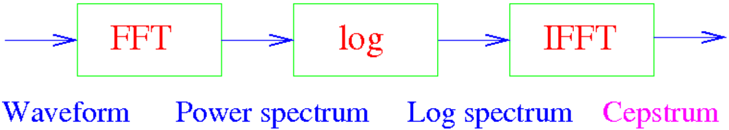
source: http://www.haskins.yale.edu/haskins/HEADS/MMSP/acoustic.html

# Speech Spectra of /th/ and /i/ sounds



Frequency: 0 Hz, amplitude: 0.2 dB

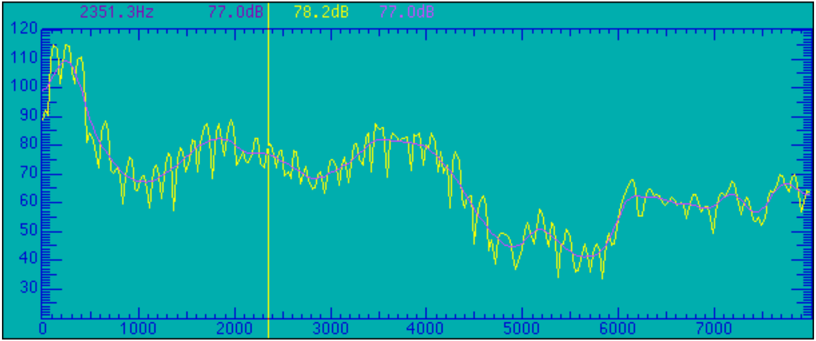
# Cepstral Analysis



FFT → log → IFFT

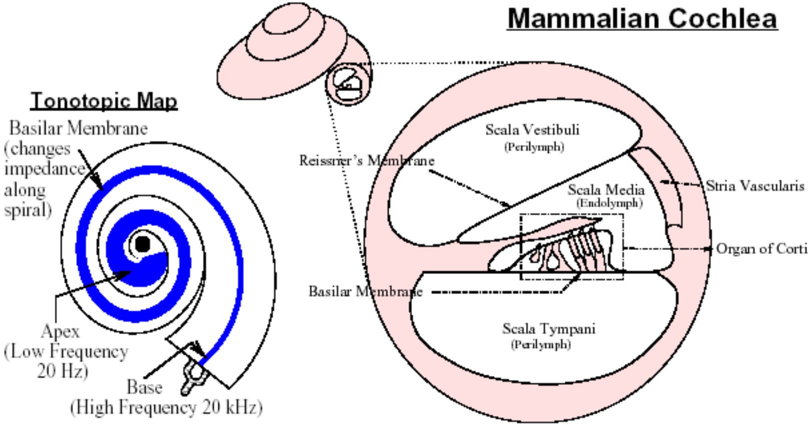
Waveform   Power spectrum   Log spectrum   Cepstrum

$$cep(q) = IFFT\{log(|S(k)| **2)\} \qquad q = 0, 1, \ldots N-1$$



Captures not only resonances but also anti-resonances.

# Hint from biology


Mammalian Cochlea
Tonotopic Map
Basilar Membrane (changes impedance along spiral)
Apex (Low Frequency 20 Hz)
Base (High Frequency 20 kHz)
Reissner's Membrane
Scala Vestibuli (Perilymph)
Scala Media (Endolymph)
Stria Vascularis
Organ of Corti
Basilar Membrane
Scala Tympani (Perilymph)

# Basilar membrane: Bark/mel scale

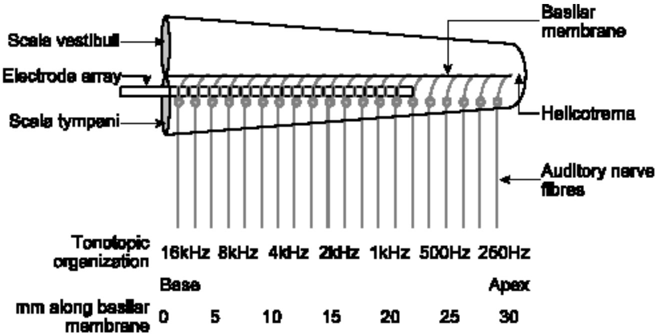
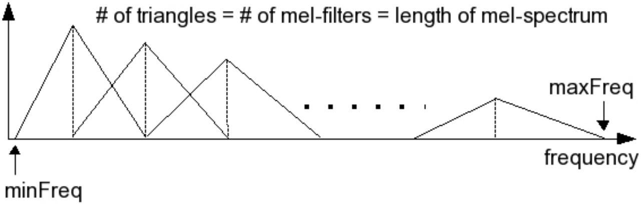


Figure 1.1. A simplified unrolled representation of the cochlea showing the auditory nerve fibres, the tonotopic organization of these nerve fibres and an intracochlear electrode array in the scala tympani.

Critical band phonomenon

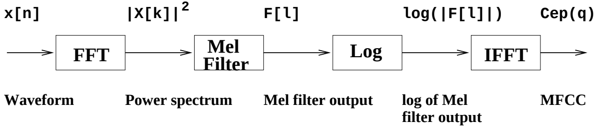
Non-linearities along amplitude and frequency

# MFCC



$$B(m) = \sum_{k=lo(m)}^{hi(m)} |X(k)|^2$$

$$cep(q) = IFFT\{log(|B(m)|^2)\} \qquad q = 0, 1, \ldots N$$



x[n] → FFT → $|X[k]|^2$ → Mel Filter → F[l] → Log → log(|F[l]|) → IFFT → Cep(q)

Waveform

Power spectrum

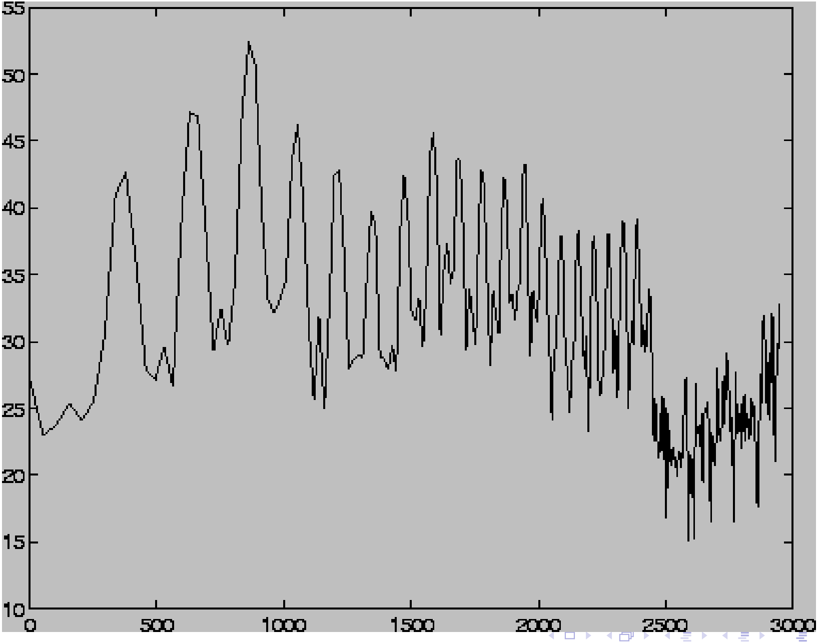
Mel filter output

log of Mel filter output

MFCC

Mel Frequency Cepstral Coefficients

# Log power spectrum after mel scale warping



# Phones and Phonemes

Phone: A sound generated by human vocal apparatus and used for human communication in a language.

Phoneme: Smallest meaningful contrastive unit in the phonology of a language.

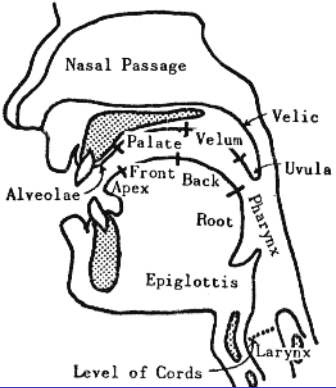
Allophones: “p” and “ph” are allophones of one phoneme /p/ in English,

are two distinct phonemes in Hindi

Minimal pair:

पल Vs फल

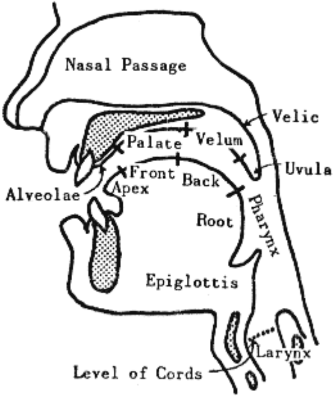
Place and Manner of articulation



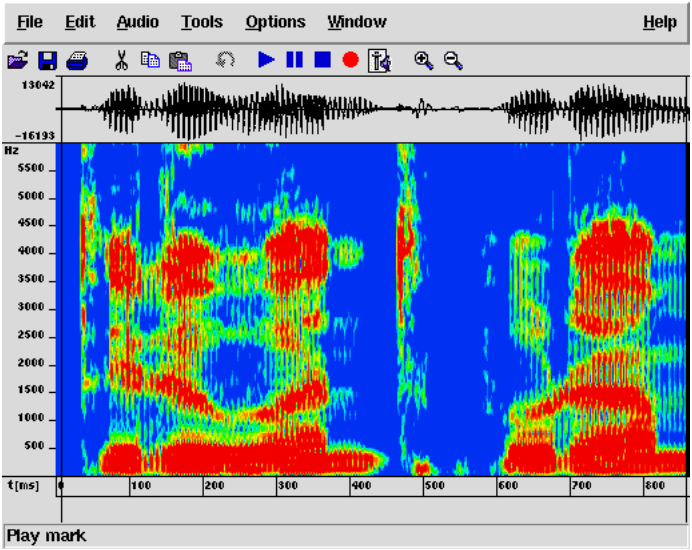
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| अ | आ | इ | ई | उ | ऊ | ए | ऐ | ओ | औ |
| *a* | *A* | *i* | *I* | *u* | *U* | *e* | *E* | *o* | *O* |

| | | | | |
|---|---|---|---|---|
| क | ख | ग | घ | ङ |
| *k* | *kh* | *g* | *gh* | *ng* |
| च | छ | ज | झ | ञ |
| *c* | *ch* | *j* | *jh* | *nj* |
| ट | ठ | ड | ढ | ण |
| *T* | *Th* | *D* | *Dh* | *N* |
| त | थ | द | ध | न |
| *t* | *th* | *d* | *dh* | *n* |
| प | फ | ब | भ | म |
| *p* | *ph* | *b* | *bh* | *m* |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| य | र | ल | व | श | ष | स | ह |
| *y* | *r* | *l* | *w* | *sh* | *S* | *s* | *h* |



# Speech: a dynamic signal



**Formant**: frequency of resonance: F1, F2, F3, ...

Slope and curvature of trajectory

# Temporal Modelling

**Delta coefficients:** $y(x) \; = \; m\,x + c$

If $Cep(n, l)$ is the $n^{th}$ cepstral coefficient at time (frame) index $l$, we can define

$$\Delta\, Cep(n, l) \; = \; \frac{\sum_{l=-L}^{L} \; l \; Cep(n, l)}{\sum_{l=-L}^{L} \; l^2}$$

**Delta-delta** (acceleration) coefficients

$$\Delta^2\, Cep(n, l) \; = \; \frac{\sum_{l=-L}^{L} \; l \; \Delta\, Cep(n, l)}{\sum_{l=-L}^{L} \; l^2}$$

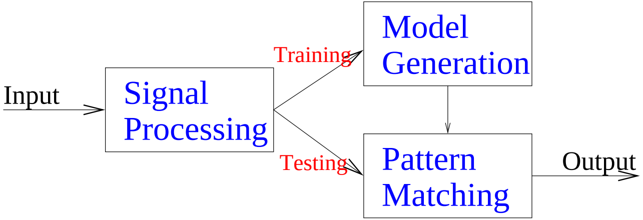
Speech signal ⇒ Sequence of feature vectors

# Speech Signal Processing (Feature Extraction)

- Digitisation of analog speech signal
- Blocking signal into frames
- FFT $\rightarrow$ mel filter $\rightarrow$ log $\rightarrow$ IFFT $\Rightarrow$ MFCC
- Slope and curvature
- Sequence of feature vectors : $x_1, x_2, \ldots x_T$

  : $o_1, o_2, \ldots o_T$

# Recognition of (static) patterns

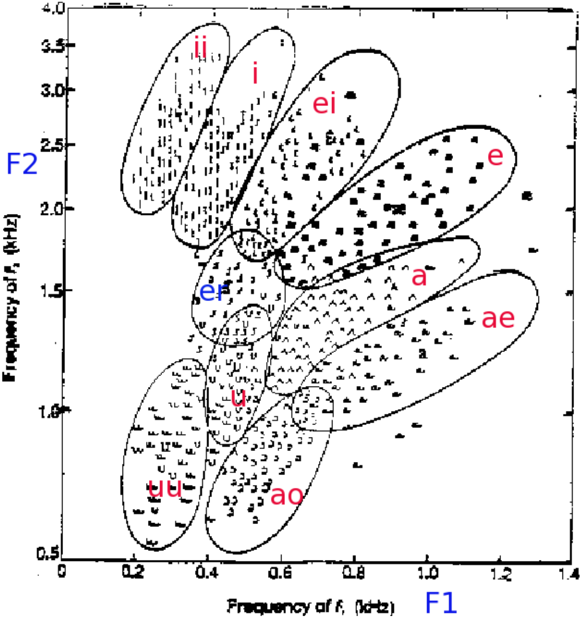


Signal Processing ⇒ Sequence of feature vectors

## Pattern Recognition

Illustration: Vowel recognition with the first 2 Formant frequencies as features

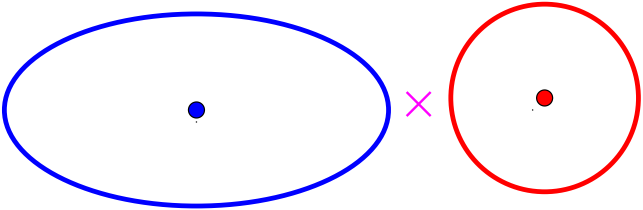
# Formant space of vowels



# Classification criterion

- *E*uclidean Distance

$$x \in C_k \quad \text{if } (x-\mu_k)^2 \;\le\; (x-\mu_j)^2 \;\; \forall j$$



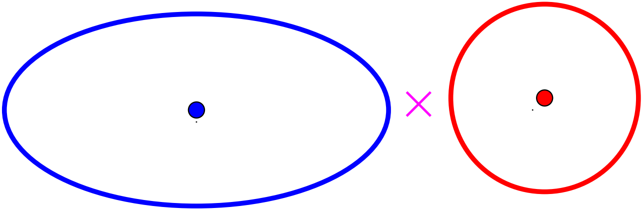
- Weighted Euclidean distance

$$d^k = \sqrt{\left(\frac{\mathbf{x}-\mu^{\mathbf{k}}}{\sigma^k}\right)^2}$$

# Classification criterion

- *E*uclidean Distance

$$x \in C_k \quad \text{if } (x-\mu_k)^2 \; \leq \; (x-\mu_j)^2 \;\; \forall j$$



- Weighted Euclidean distance

$$d^k = \sqrt{\left(\frac{\mathbf{x}-\mu^{\mathbf{k}}}{\sigma^k}\right)^2}$$

- Extension to multiple features

$$d^k = \sqrt{\sum_i \left(\frac{\mathbf{x_i}-\mu_{\mathbf{i}}^{\mathbf{k}}}{\sigma_i^k}\right)^2}$$

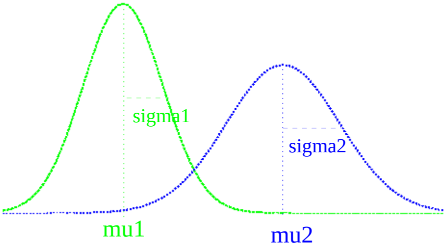
$d(\overline{\mathbf{x}}, \overline{\mu_{\mathbf{k}}})$

Probabilistic models

# Two class problem

Normal Distribution: $N(\mu;\sigma)$

$$p(x) \;=\; \frac{1}{\sqrt{2\pi\sigma^2}} exp\left\{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right\}$$

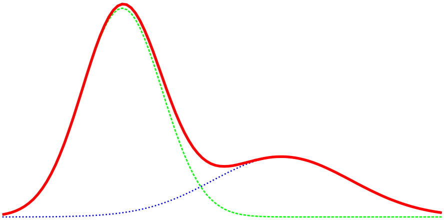


Maximum Likelihood classification criterion:

$$x \in C_k \quad \text{if } p(x|N(\mu_k;\sigma_k)) \;\geq\; p(x|N(\mu_j;\sigma_j)) \quad \forall j$$

Refer to vowel F1-F2 diagram

# Gaussian Mixture Model(GMM)



$$p(x|GMM(k)) \;= \alpha p(x : N[\mu_1;\sigma_1]) \quad + (1-\alpha)\; p(x : N[\mu_2;\sigma_2])$$

Maximum Likelihood classification criterion for GMM case:

$$x \in C_k \quad \text{if } p(x|GMM(k)) \;\geq\; p(x|GMM(j)) \quad \forall j$$

Extension to Multi-dimensional space

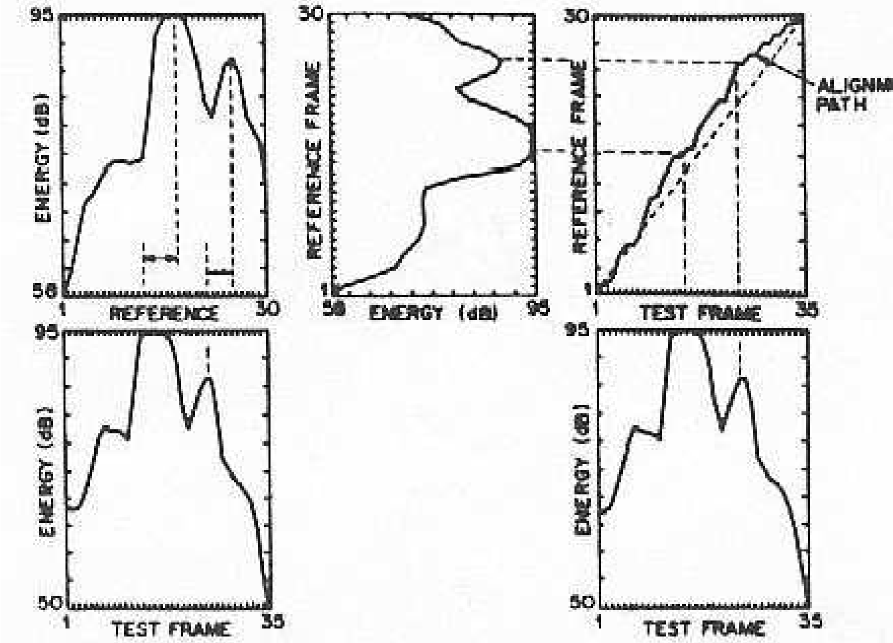
# Classification of Temporal patterns

Isolated Word Recognition:
Example: name dialling

Match a sequence of test feature vectors $x_1, x_2, \ldots, x_N$
with a sequence of reference feature vectors $r_1, r_2, \ldots, r_M$
Reasons for $N \neq M$

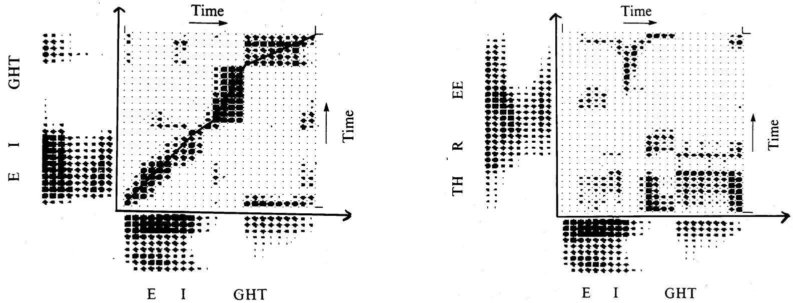
- End-point detection errors
- speaking rate variations
- Within word variations

Linear vs Non-linear Time-warping



From: Fundamentals of Speech Recognition, L.Rabiner and B.H.Juang

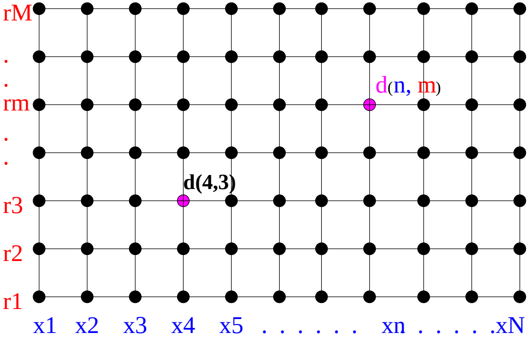
# Optimal alignment path



From: Holmes book

Bigger the dark blob, greater the similarity (lesser distance).
"eight" versus "eight": A path along diagonal exists
"eight" versus "three": A path along diagonal does not exist.

# Dynamic Programming

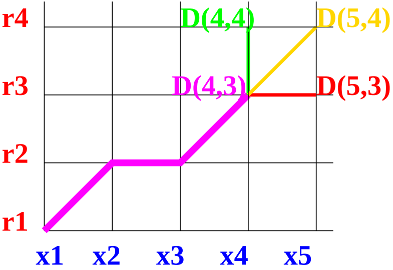


Test feature vector sequence

Goal: To find the optimal alignment path from the grid point $(1, 1)$ to the grid point $(N, M)$. There are exponential number $(M^N)$ of paths. In order to reduce the number of computations from exponential to linear, we use the Dynamic Programming whose foundation is the “principle of optimality”.

**Principle of optimality**: The best path from $(1, 1)$ to any given point on the grid is independent of what happens beyond that point.

So, if two paths share a partial path starting from $(1, 1)$, the cost of this shared partial path need to be computed only once and stored in a table for later use.



DP Algorithm: Define

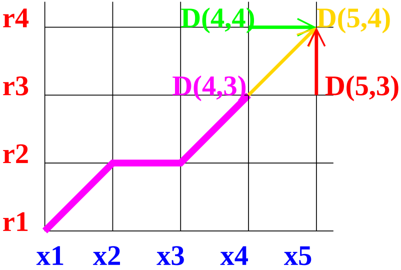
$d(n, m)$ : the **local** distance between the $n^{th}$ test frame and $m^{th}$ reference frame.

$D(n, m)$ : the **accumulated** distance of the optimal path starting from the grid point $(1, 1)$ and ending at the grid point $(n, m)$.

# Dynamic Time Warping

Applying the *P*rinciple of optimality, $D(n,m)$ is the sum of the local cost, and the cost of cheapest path to it

$$\mathbf{D(5,4) = d(5,4) + min} \begin{cases} \mathbf{D(4,4)} \\ \mathbf{D(4,3)} \\ \mathbf{D(5,3)} \end{cases}$$

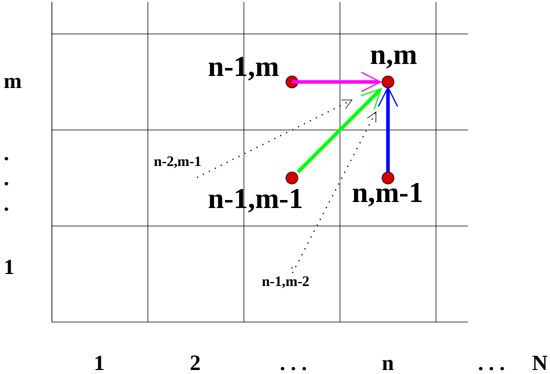


$$D(n,m) = d(n,m) + min \begin{cases} D(n-1,m) \\ D(n-1,m-1) \\ D(n,m-1) \end{cases}$$

$*$ Compute $D(n,m)$ for each "allowed" pair of $(n,m)$. Remember the "best" predecessor point.

$*$ $D(N,M)$ is the cost of the optimal path.

$*$ From $(N,M)$, start backtracing to identify the optimal path.



$$D(n,m) = d(n,m) + min \begin{cases} D(n-1,m) \\ D(n-1,m-1) \\ D(n,m-1) \end{cases}$$

$*$ Compute $D(n,m)$ for each "allowed" pair of $(n,m)$. Remember the "best" predecessor point.

$*$ $D(N,M)$ is the cost of the optimal path.

$*$ From $(N,M)$, start backtracing to identify the optimal path.

Global constraints: left- and down-paths are prohibited.

Local constraints: path $(n,m-1) \rightarrow (n,m)$ not allowed.

# Spell checking: Application of Dynamic Programming

Reference (correct spelling)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| n | 1 | 1 | 2 | | | 1 | 1 | 0 |
| r | 1 | 1 | 2 | 1 | 2 | 0 | 0 | 1 |
| e | 2 | 2 | 1 | 2 | 1 | 2 | 2 | 2 |
| t | 1 | 1 | 2 | 0 | 2 | 1 | 1 | |
| t | 1 | 1 | 2 | 0 | 2 | 1 | 1 | |
| a | 2 | 2 | 0 | 2 | 0 | 2 | | |
| p | 0 | 0 | 2 | | 2 | 1 | | |
| | p | p | a | t | a | r | r | n |

Test sequence (just typed in text)

d(v,c)=2

d(v1,v2)=1

d(c1,c2)=1

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| n | 1 8 | 1 | 2 | | | 1 2 | 1 2 | 0 1 |
| r | 1 7 | 1 | 2 | 1 | 2 3 | 0 1 | 0 1 | 1 2 |
| e | 2 6 | 2 | 1 | 2 2 | 1 1 | 2 3 | 2 | 2 |
| t | 1 4 | 1 4 | 2 4 | 0 0 | 2 2 | 1 | 1 | |
| t | 1 3 | 1 3 | 2 2 | 0 0 | 2 2 | 1 | 1 | |
| a | 2 2 | 2 2 | 0 0 | 2 2 | 0 | 2 | | |
| p | 0 0 | 0 0 | 2 2 | 2 4 | 2 | 1 | | |
| | p | p | a | t | a | r | r | n |

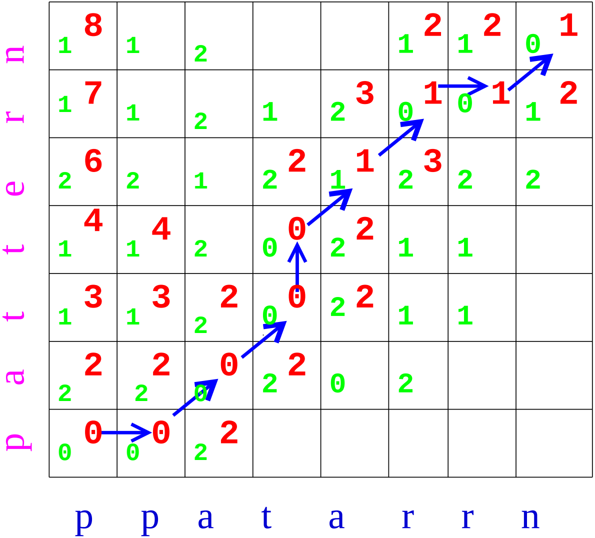
Test sequence (just typed in text)

$d(v,c)=2$

$d(v1,v2)=1$

$d(c1,c2)=1$

$$D(x,y)= d(x,y) + \min \begin{cases} D(x-1,y-1) \\ D(x-1,y) \\ D(x,y-1) \end{cases}$$



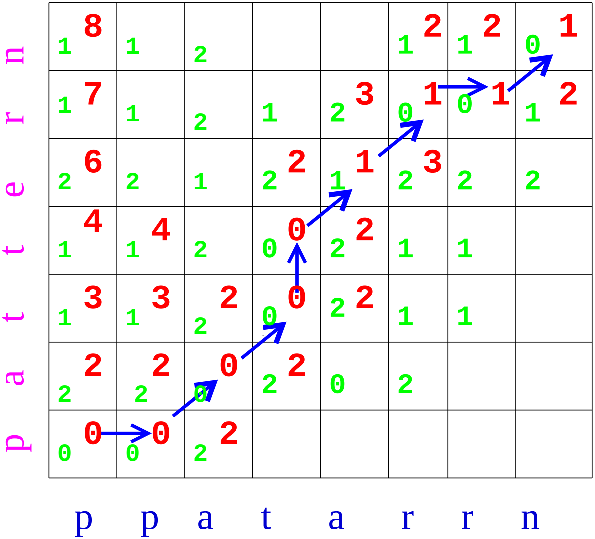
|  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|
| n | 1 8 | 1 | 2 |  |  | 1 2 | 1 2 | 0 1 |
| r | 1 7 | 1 | 2 | 1 | 2 3 | 0 1 | 0 1 | 1 2 |
| e | 2 6 | 2 | 1 | 2 2 | 1 1 | 2 3 | 2 | 2 |
| t | 1 4 | 1 4 | 2 | 0 0 | 2 2 | 1 | 1 |  |
| t | 1 3 | 1 3 | 2 2 | 0 0 | 2 2 | 1 | 1 |  |
| a | 2 2 | 2 2 | 0 0 | 2 2 | 0 | 2 |  |  |
| p | 0 0 | 0 0 | 2 2 |  |  |  |  |  |
|  | p | p | a | t | a | r | r | n |

d(v,c)=2

d(v1,v2)=1

d(c1,c2)=1

$$D(x,y) = d(x,y) + \min \begin{cases} D(x-1,y-1 \\ D(x-1,y) \\ D(x,y-1) \end{cases}$$



|  | p | p | a | t | a | r | r | n |
|---|---|---|---|---|---|---|---|---|
| n | 1 8 | 1 | 2 |  |  | 1 2 | 1 2 | 0 1 |
| r | 1 7 | 1 | 2 | 1 | 2 3 | 0 1 | 0 1 | 1 2 |
| e | 2 6 | 2 | 1 | 2 2 | 1 1 | 2 3 | 2 | 2 |
| t | 1 4 | 1 4 | 2 | 0 0 | 2 2 | 1 | 1 |  |
| t | 1 3 | 1 3 | 2 2 | 0 0 | 2 2 | 1 | 1 |  |
| a | 2 2 | 2 2 | 0 0 | 2 2 | 0 | 2 |  |  |
| p | 0 0 | 0 0 | 2 2 |  |  |  |  |  |

$d(v,c)=2$

$d(v1,v2)=1$

$d(c1,c2)=1$

$$D(x,y)= d(x,y) + \min \begin{cases} D(x-1,y-1) \\ D(x-1,y) \\ D(x,y-1) \end{cases}$$

Training: Viterbi (forced) alignment to get phoneme boundaries

Reference template generation: average frames belonging same phoneme

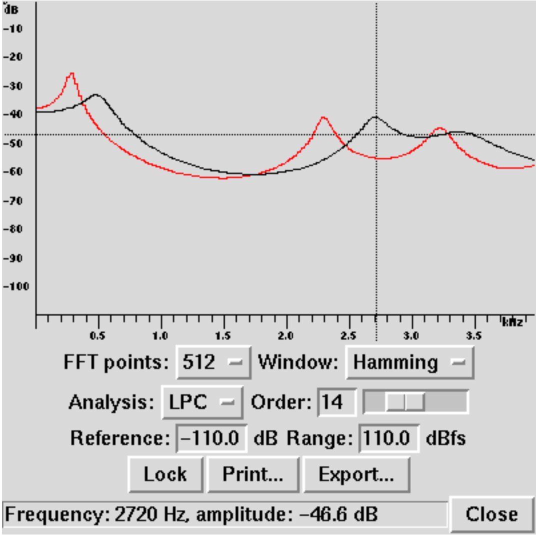
Recognition: Viterbi traceback to retrieve phoneme sequence

# Why speech recognition is difficult?

Sources of variabilities

- Speaker specific: physiological, emotional, cultural
- Continuous signal: no well defined boundaries between linguistic units
- Ambience: noise, Lombard effect, room acoustics
- Channel: additive/convolutional noise, compression
- Transducer: omni/uni-directional, carbon/electret mic
- Phonetic context

## Spectra of the vowel ‘i’ in word “pin” spoken by male and female speakers



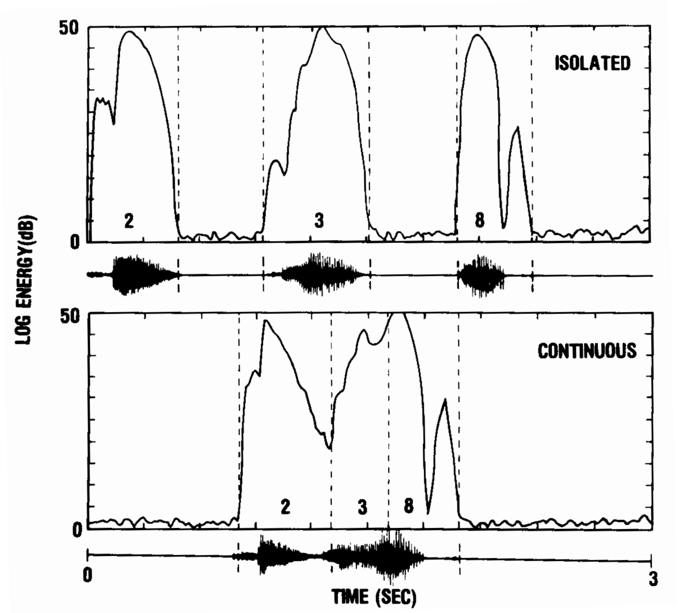
FFT points: 512 Window: Hamming

Analysis: LPC Order: 14

Reference: −110.0 dB Range: 110.0 dBfs

Frequency: 2720 Hz, amplitude: −46.6 dB

# No well defined boundaries between linguistic units



# Diversity of transduction characteristics of microphones

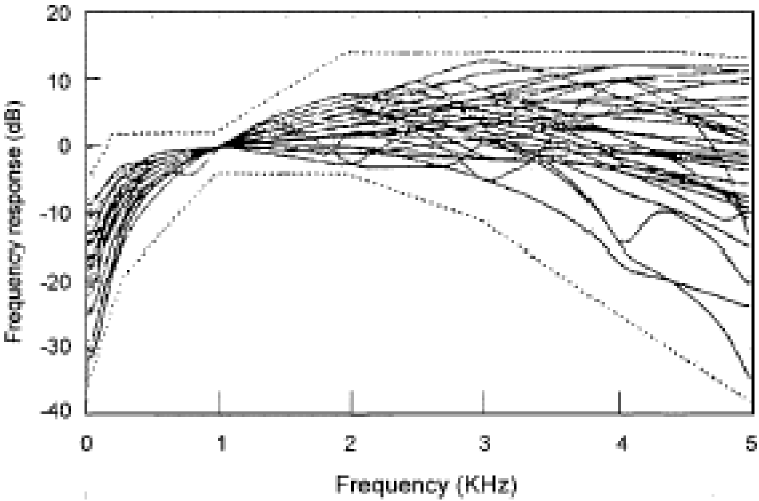
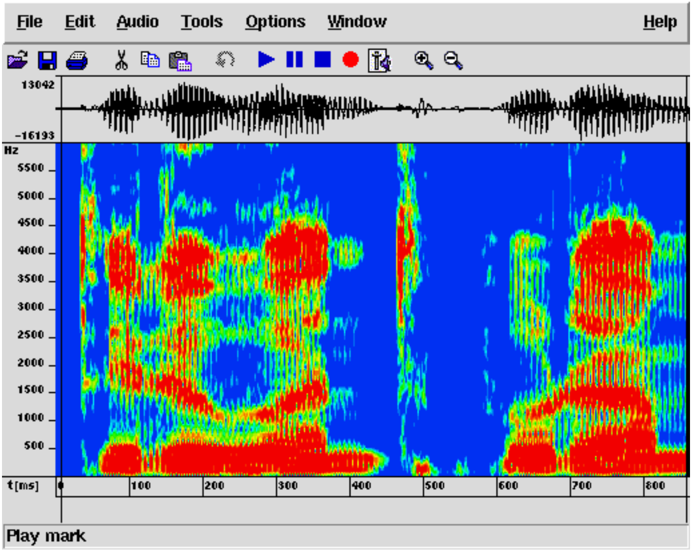


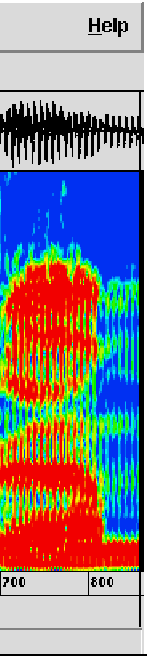
Fig. 6. Diversity of transducer characteristics in telephone set [25].

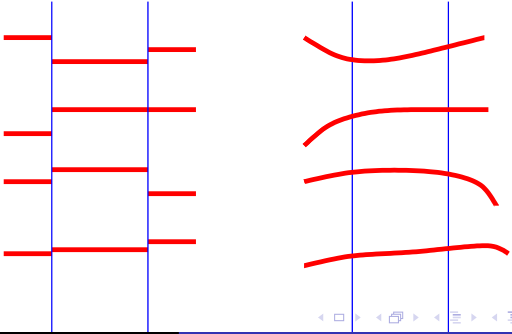
# Spectrogram of thiruvananthapuram



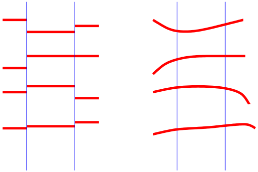
t i r u w a n th p u r a m

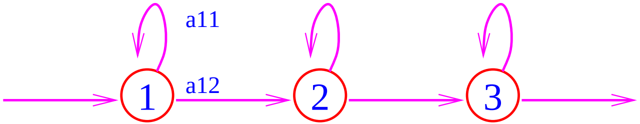
# Formant trajectories

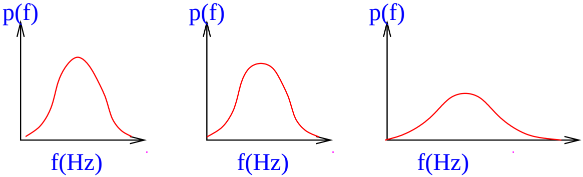




# hidden Markov model (HMM)

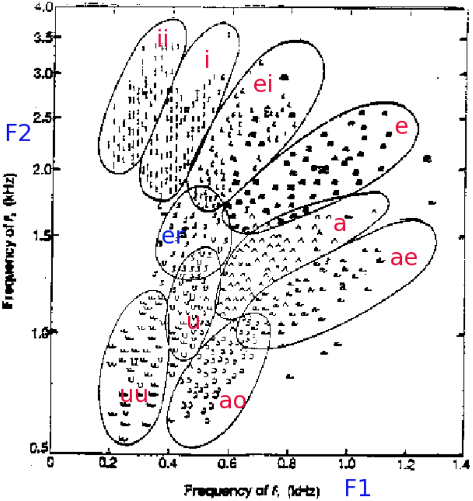






Parameters of a HMM: A, B, $\pi$

# What is hidden in hidden Markov model?



# 3 problems in HMM

- How to compute the likelihood of a trained model generating a test observation sequence?
  Solution: forward algorithm (uses DP)
- How to find the optimal state sequence?
  Solution: Viterbi algorithm (similar to DTW)
- How to estimate the parameters of the model: $\lambda = (A, B, \pi)$?
  Solution: Forward-backward (Baum-Welch) algorithm

# DP and HMM: Viterbi algorithm

In case of template matching (DTW), we decided on the optimal path that minimised distance between a test feature sequence and a reference template. The key optimisation equation was

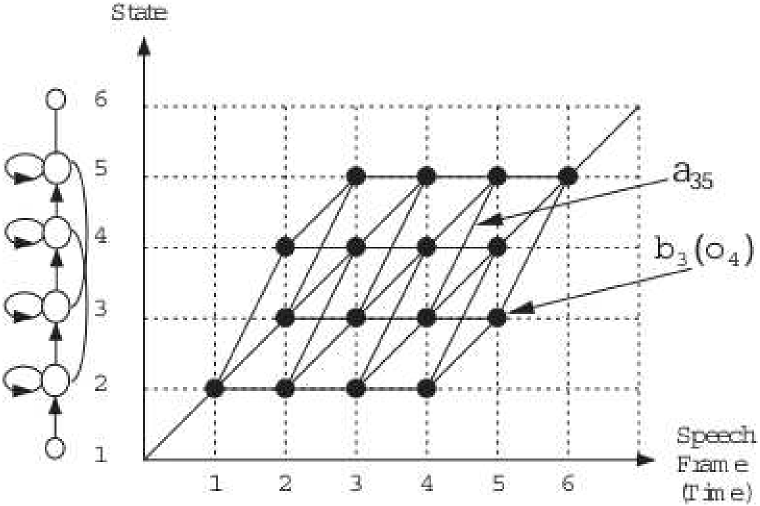
$$D(n,m) = d(n,m) + min \begin{cases} D(n-1,m) \\ D(n-1,m-1) \\ D(n,m-1) \end{cases}$$

In case of a probabilistic model, we want to maximise the probability of a test feature sequence matching a HMM.
In the log probability domain, the DP equation for matching a test sequence with the best HMM state sequence (Viterbi algorithm) is

$$\psi_j(t) = log(b_j(\mathbf{o}_t)) + \max_i\{\psi_i(t-1) + log(a_{ij})\}$$

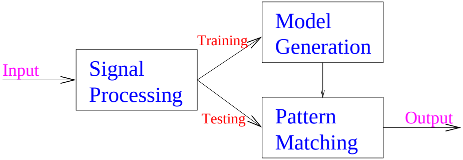
Initial conditions: $\psi_1(1) = 0; \psi_j(1) = log(a_{1j}) + log(b_j(\mathbf{o}_1))$

# DP and HMM: Viterbi algorithm



source: The HTK Book

The HMM can represent even a sentence!



Input

Signal Processing

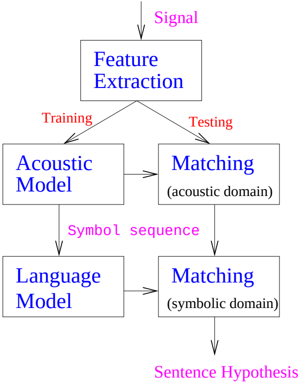
Training

Testing

Model Generation

Pattern Matching

Output



Signal

Feature Extraction

Training

Testing

Acoustic Model

Matching (acoustic domain)

Symbol sequence

Language Model

Matching (symbolic domain)

Sentence Hypothesis

Recognition of a spoken sentence (a sequence of words)

# Knowledge sources

Phone sequence/phone hypothesis lattice

==> Sentence hypothesis

Lexicon

man

mna

Syntax

Some man brought the apple.

Apple the brought man some.

# Knowledge sources

Phone sequence/phone hypothesis lattice

==> Sentence hypothesis

Lexicon

man

mna

Syntax

Some man brought the apple.

Apple the brought man some.

Semantics

Time flies like an arrow

Fruit flies like banana

Pragmatics

Turn left for the nearest chemist

# Combining Acoustic and Language Models

Let $Y$ : Acoustic feature sequence

$W$ : Word sequence

$$\widehat{\mathbf{W}} = \underset{\mathbf{W}}{argmax} \quad P(\mathbf{W}|\mathbf{Y})$$

# Combining Acoustic and Language Models

Let $Y$ : Acoustic feature sequence

$W$ : Word sequence

$$\widehat{\mathbf{W}} = \underset{\mathbf{W}}{argmax} \quad P(\mathbf{W}|\mathbf{Y})$$

Bayes' rule:

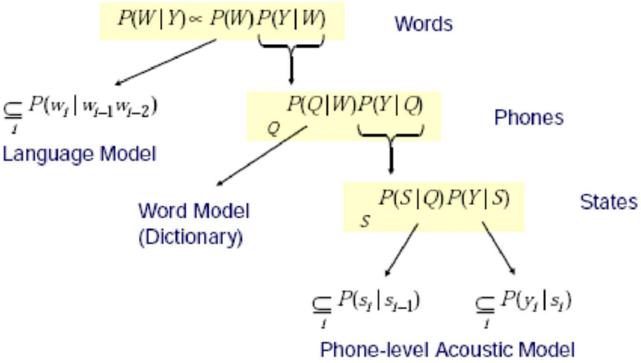
$$P(\mathbf{W}|\mathbf{Y}) = \frac{P(\mathbf{Y}|\mathbf{W})P(\mathbf{W})}{P(\mathbf{Y})}$$

$$\widehat{\mathbf{W}} = \underset{\mathbf{W}}{argmax} \quad \frac{P(\mathbf{Y}|\mathbf{W})P(\mathbf{W})}{P(\mathbf{Y})}$$

CSR: Acoustic model, Language model and Hypothesis search

# Hierarchy of Units

$$\widehat{\mathbf{W}} = \underset{\mathbf{W}}{argmax} \quad \frac{P(\mathbf{Y}|\mathbf{W})P(\mathbf{W})}{P(\mathbf{Y})}$$



$P(W|Y) \propto P(W)\underbrace{P(Y|W)}$ Words

$\underset{t}{\subseteq} P(w_t \mid w_{t-1} w_{t-2})$

Language Model

$\underset{Q}{} P(Q|W)\underbrace{P(Y|Q)}$ Phones

Word Model (Dictionary)

$\underset{S}{} P(S|Q)P(Y|S)$ States

$\underset{t}{\subseteq} P(s_t \mid s_{t-1})$ $\quad \underset{t}{\subseteq} P(y_t \mid s_t)$

Phone-level Acoustic Model

"Beads on a string model"

Source: "State of the Art in ASR (and beyond)", Steve Young

# Basic units of HMM (phone-like units)

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| अ | आ | इ | ई | उ | ऊ | ए | ऐ | ओ | औ |
| $a$ | $A$ | $i$ | $I$ | $u$ | $U$ | $e$ | $E$ | $o$ | $O$ |

| | | | | |
|---|---|---|---|---|
| क | ख | ग | घ | ङ |
| $k$ | $kh$ | $g$ | $gh$ | $ng$ |
| च | छ | ज | झ | ञ |
| $c$ | $ch$ | $j$ | $jh$ | $nj$ |
| ट | ठ | ड | ढ | ण |
| $T$ | $Th$ | $D$ | $Dh$ | $N$ |
| त | थ | द | ध | न |
| $t$ | $th$ | $d$ | $dh$ | $n$ |
| प | फ | ब | भ | म |
| $p$ | $ph$ | $b$ | $bh$ | $m$ |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| य | र | ल | व | श | ष | स | ह |
| $y$ | $r$ | $l$ | $w$ | $sh$ | $S$ | $s$ | $h$ |

# Pronunciation dictionary

* Representing a word as a sequence of units of recognition

* Pronunciation rules can be used

* Manual verification is necessary

kalam vs kamal

karnaa, pahale, Bhaartiya

pause

| | |
|---|---|
| aage | aa g e |
| aaja | aa j |
| aba | a b |
| abbaasa | a bb aa s |
| aatxha | aa t'h |

# Pronunciation dictionary

* Representing a word as a sequence of units of recognition
* Pronunciation rules can be used
* Manual verification is necessary

kalam vs kamal
karnaa, pahale, Bhaartiya
pause

| | |
|---|---|
| aage | aa g e |
| aaja | aa j |
| aba | a b |
| abbaasa | a bb aa s |
| aatxha | aa t’h |

Multiple pronunciations

| | |
|---|---|
| vij~nAna | v i j n aa n |
| vij~nAna(2) | v i g y aa n |

# Examples of pronunciation variability

Feature spreading in coalescence:

c ae n t − > c ae t where ae is nasalised

Assimilation causing changes in place of articulation:

n − > m before labial stop as in input, can be, grampa

Asynchronous articulation errors causing stop insertions:

warm[p]th, ten[t]th, on[t]ce, leng[k]th

Fast speech:

probably − − > probly

r-insertion in vowel-vowel transitions:

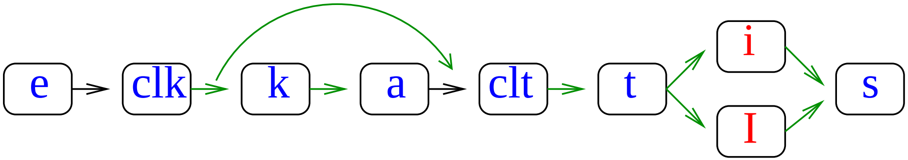
stir [r]up, director [r]of

Context dependent deletion:

nex[t] week

Source: "State of the Art in ASR (and beyond)", Steve Young

# phone HMM → word HMM → sentence HMM



e clk k a clt t I s        e clk clt t I s

e clk k a clt t i s        e clk clt t i s

* "probabilities" of pronunciations can be estimated

* many pronunciations → higher word confusions

→ performance degradation

* Dialect and Accent (native/non-native speakers)

* seek a dynamic speaker specific pron dictionary.

# Training subword HMMs

An iterative algorithm (Baum-Welch, also known as Forward-Backward) is used. The Maximum Likelihood approach guarantees increase of the likelihood of the trained model matching with training data with each iteration. To begin with, an initial estimation of parameters of HMMs $(A, B, \pi)$ is required.

Q: How to get an initial estimation of $(\lambda = \{A, B, \pi\}$?

A: We can estimate parameters if we know the boundaries of every subword HMM in training utterances.

# Training subword HMMs

An iterative algorithm (Baum-Welch, also known as Forward-Backward) is used. The Maximum Likelihood approach guarantees increase of the likelihood of the trained model matching with training data with each iteration. To begin with, an initial estimation of parameters of HMMs $(A, B, \pi)$ is required.

Q: How to get an initial estimation of $(\lambda = \{A, B, \pi\}$?

A: We can estimate parameters if we know the boundaries of every subword HMM in training utterances.

Practical solution: Assume that the durations of all units (phones) are equal. If there are $N$ phones in a training utterance, divide the feature vector sequence into $N$ equal parts. Assign each part, to a phoneme in the phoneme sequence corresponding to the transcription of the utterance. Repeat for all training utterances.

# Initial estimation of HMM parameters: an illustration

Let the transcription of the 1st wave file be the following sequence of words: mera bhaarat mahaan

Let the relevant lines in the dictionary be as follows:

| | |
|---|---|
| bhaarata | bh aa r a t |
| mahaana | m a h aa n |
| mera | m e r aa |

The phonemeHMM sequence (of length 16) corresponding to this sentence is sil m e r aa bh aa r a t m a h aa n sil

# Initial estimation of HMM parameters: an illustration

Let the transcription of the 1st wave file be the following sequence of words: mera bhaarat mahaan

Let the relevant lines in the dictionary be as follows:

| | |
|---|---|
| bhaarata | bh aa r a t |
| mahaana | m a h aa n |
| mera | m e r aa |

The phonemeHMM sequence (of length 16) corresponding to this sentence is sil m e r aa bh aa r a t m a h aa n sil

If the duration of the wavefile is 1.0sec, there will 98 feature vectors (frame shift = 10msec and frame size = 25msec).

Assign the first 6 feature vectors to "sil" HMM; the next 6 (7 through 12) to "m"; the next 6 (13 through 18) to "e"; ... ; the last 8 feature vectors to "sil". If HMM has 3 states, assign 2 feature vector to each state; compute mean,SD.
Assume $a_{i,j}$=0.5 if j=i or j=i+1; else assign 0.

# Decoding: Generation of word hypotheses

Generation of word hypotheses can result in

- a single sequence of words,
- in a collection of the n-best word sequences,
- in a lattice of partially overlapping word hypotheses.

# Decoding: Generation of word hypotheses

Generation of word hypotheses can result in

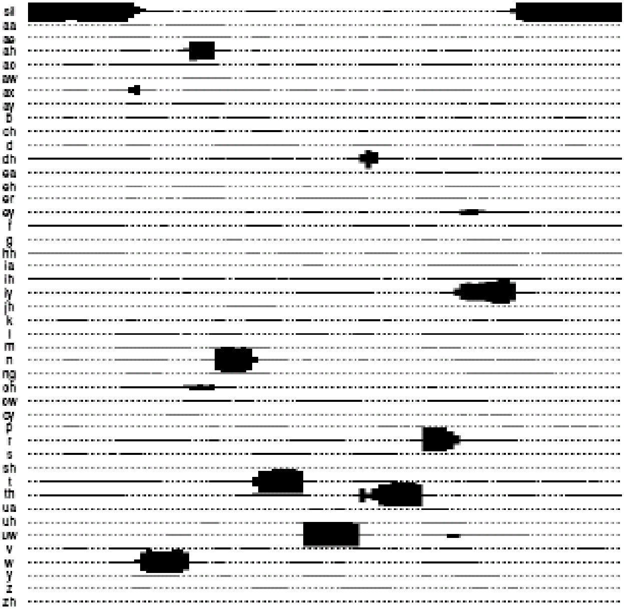
- * a single sequence of words,
- * in a collection of the n-best word sequences,
- * in a lattice of partially overlapping word hypotheses.

Goal: Find the path with the least cost (most likely word sequence)

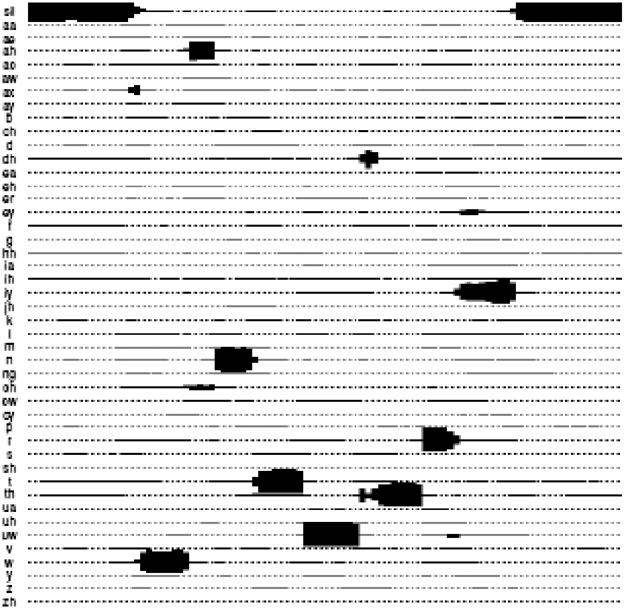
Acoustic evidence → Word lattice $--->$ DAG

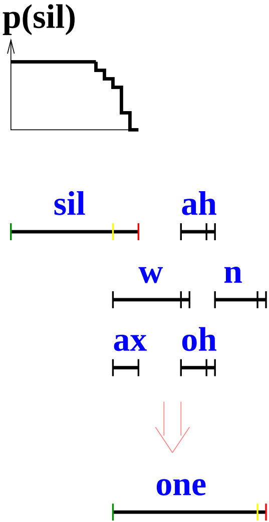
Given a graph with N nodes and E edges, the least-cost path can be found in time proportional to N+E

# Probabilities of phones at various time instants

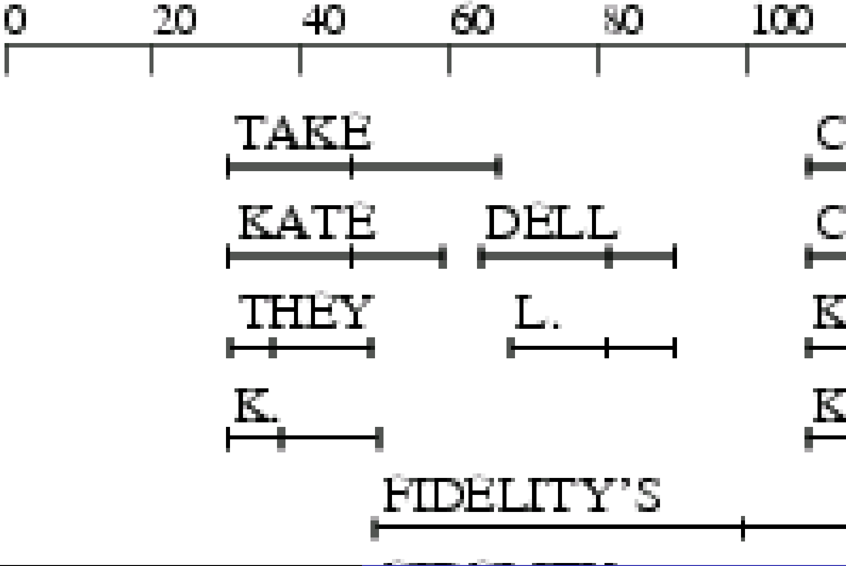


# Probabilities of phones at various time instants

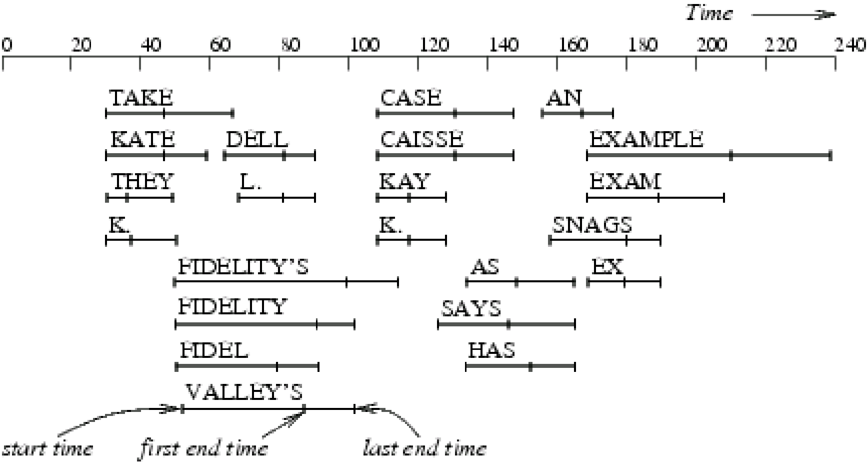




# Lattice of phone hypotheses → lattice of word hypotheses



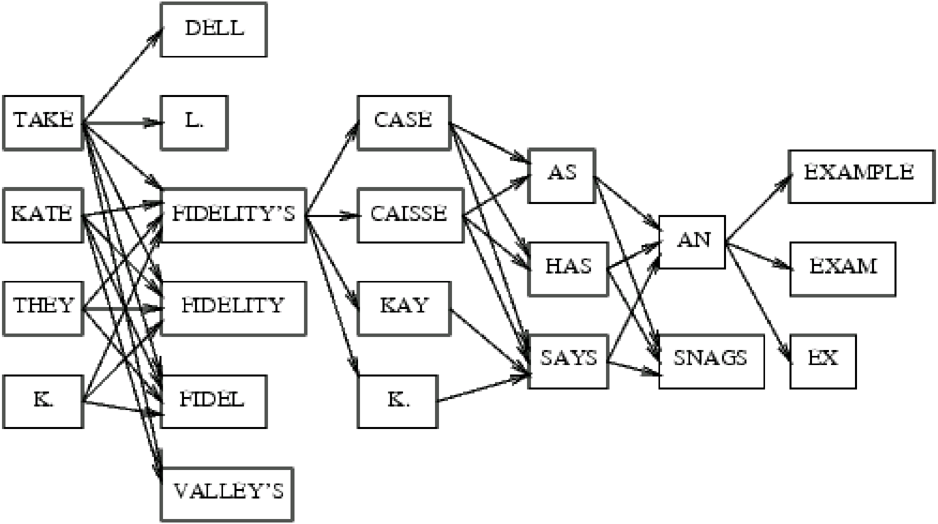
# Word hypotheses at various time instants



Take Fidelity's case as an example

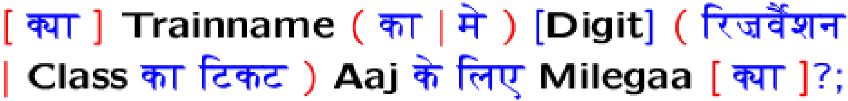
Source: "Efficient algorithms for Speech Recognition", M.K.Ravishankar, PhD thesis: CMU-CS-96-143

# Word Lattice as a Directed Acyclic Graph



# Incorporation of syntax

Backus-Naur Form (BNF) grammar is useful for ASR in a specific task domain.



[ क्या ] Trainname ( का | मे ) [Digit] ( रिजर्वैंशन | Class का टिकट ) Aaj के लिए Milegaa [ क्या ]?;

Integration of syntax, semantics and domain knowledge

# Statistical model: n-grams

## Probability of a word sequence

Let **W** denote the word sequence $w_1, w_2, \cdots, w_i$.

$$p(\mathbf{W}) = p(w_1) \times p(w_2|w_1) \times p(w_3|w_1, w_2) \times \cdots \times p(w_i \mid w_{i-1}, w_{i-2}, \cdots, w_1)$$

Not practical due to 'unlimited history':
too many parameters for even a short **W**

## Markovian assumption:

- Disregard 'very old' history (short memory)
- remember only 'n-1' previous words: n-gram model

# Parameter Estimation

Maximum Likelihood Estimation: relative frequencies
Use counts from training data.

unigram:

$$p(w) = C(w)/|V|$$

# Parameter Estimation

Maximum Likelihood Estimation: relative frequencies
Use counts from training data.

unigram:

$$p(w) = C(w)/|V|$$

bigram:

$$p(w_n|w_{n-1}) = \frac{C(w_{n-1}, w_n)}{\sum_w C(w_{n-1} w_n)}$$

$$p(w_n|w_{n-1}) = \frac{C(w_{n-1}, w_n)}{C(w_{n-1})}$$

# Parameter Estimation

Maximum Likelihood Estimation: relative frequencies
Use counts from training data.

unigram:

$$p(w) = C(w)/|V|$$

bigram:

$$p(w_n|w_{n-1}) = \frac{C(w_{n-1}, w_n)}{\sum_w C(w_{n-1} w_n)}$$

$$p(w_n|w_{n-1}) = \frac{C(w_{n-1}, w_n)}{C(w_{n-1})}$$

n-gram:

$$p(w_n|w_1 w_2 \cdots w_{n-1}) = \frac{C(w_1, w_2, \cdots, w_{n-1}, w_n)}{C(w_1, w_2, \cdots, w_{n-1})}$$

# Data sparsity ⇒ Smoothing of probability distributions

Example: 1000 word vocabulary corpus divided into training set of size 1,500,000 words and test set of size 300,000 words.

Observation: 23% of the trigrams occuring in test data never occurred in the training subset!
Similar observation with a 38 million word newspaper corpus.

Robust parameter estimation is needed

## Eliminating Zero Probabilities

From the same training data, derive revised n-grams such that no n-gram is zero.

Discounting: Take away some counts from 'high count words' and distribute them among 'zero/low count words'.

# Good-Turing Discounting

Let $N_c$ denote the number of bigrams that occured $c$ times in the corpus.
For bigrams that never occured, the revised count is

$$c^* = \frac{N_1}{N_0}$$

# Good-Turing Discounting

Let $N_c$ denote the number of bigrams that occured $c$ times in the corpus.
For bigrams that never occured, the revised count is

$$c^* = \frac{N_1}{N_0}$$

In general,

$$c^* = (c+1)\frac{N_{c+1}}{N_c}$$

# Good-Turing Discounting

Let $N_c$ denote the number of bigrams that occured $c$ times in the corpus.
For bigrams that never occured, the revised count is

$$c^* = \frac{N_1}{N_0}$$

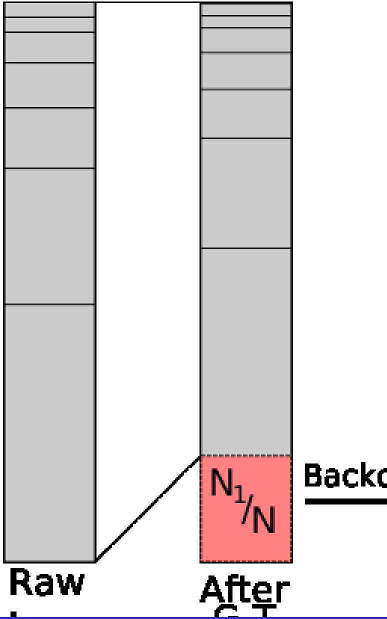
In general,

$$c^* = (c+1)\frac{N_{c+1}}{N_c}$$

* Proper normalization is needed.

* Suitable for estimation from large data.

# Good-Turing Discounting: Illustration



# Using n-gram 'hierarchy': Combining frequencies

Linear interpolation of n-grams

$$\hat{p}(w_3|w_1, w_2) = \lambda_1 p(w_3|w_1, w_2) + \lambda_2 p(w_3|w_2) + \lambda_3 p(w_3)$$

$$\text{with } \lambda_i > 0; \quad \sum_i \lambda_i = 1.0$$

# Using n-gram ‘hierarchy’: Combining frequencies

Linear interpolation of n-grams

$$\hat{p}(w_3|w_1, w_2) = \lambda_1 p(w_3|w_1, w_2) + \lambda_2 p(w_3|w_2) + \lambda_3 p(w_3)$$

$$\text{with } \lambda_i > 0; \quad \sum_i \lambda_i = 1.0$$

## Backoff trigram

if trigram count $> 0$ no interpolation
Backoff to bigram otherwise

We “backoff” to a lower order n-gram only if we have zero evidence for a higher order n-gram.

A non-linear method of combining counts.

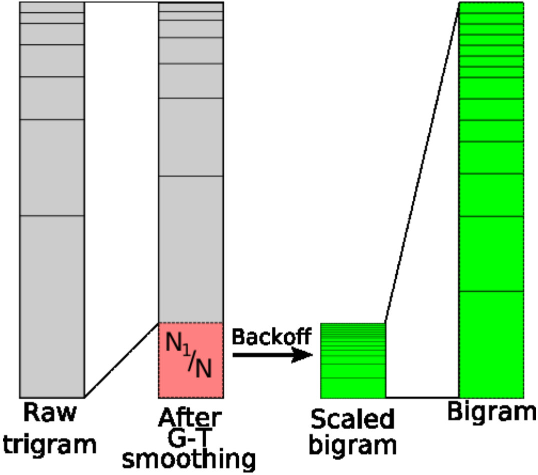
# Backoff Grammar

The backoff trigram grammer is computed as

```
if  (trigramCount(xyz)) > 0) {
                 // compute trigramProb(z|xy)
else if (bigramCount(yz)) > 0){
                 trigramProb = a1(xy) * bigramProb(z|y)
} else {
                 trigramProb = a2(y)  * unigramProb(z)
}
```

a1 and a2 are positive scale factors that can even be $>1$ (for a lucid explanation, see http://www.speech.cs.cmu.edu/sphinxman/FAQ.html).

# Backoff trigram grammar with Good-Turing Discounting



http://www.inf.ed.ac.uk/teaching/courses/fnlp/lectures/9/index_html.xml#(4)

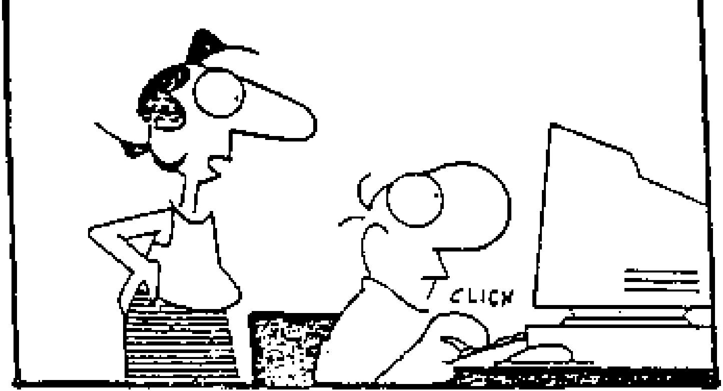
# Requirements for Implementation of an ASR system

- Knowledge of spoken language recognition
- ASR toolkit
- Speech data
- Transcription (sequence of 'words' in an utterance)
- Pronunciation dictionary
- Language model (can be generated automatically)
- Knowledge of shell scripts and perl helps
- Lots of patience and perseverance

## A Short list of Relevant Books

1. "Speech Communications : Human and Machine", D. O'Shaughnessy University press, Hyderabad; price: Rs. 575.
2. "Fundamentals of Speech Recognition", by Lawrence R. Rabiner, B. H. Juang and B.Yegnanarayana, Pearson Education India, 2008, Rs. 450; ISBN:9788177585605
3. "Statistical Methods for Speech Recognition", Frederick Jelinek, The MIT Press, 1997.
4. "Spoken Language Processing : A Guide to Theory, Algorithm and System Development", by Xuedong Huang, Alex Acero, Hsiao-Wuen Hon Year 2001, Prentice Hall PTR; ISBN: 0130226165; Price: $91.
5. "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition", By Daniel Jurafsky and J.H.Martin, ISBN 8178085941 Pearson Education Asia, 2000. Price Rs. 425
6. "Spoken Language Understanding — An Introduction to the Statistical Framework" Y. Wang, L. Deng, and A. Acero, In IEEE Signal Processing Magazine, Vol 27 No. 5. Sepetmber 2005.

More links at http://speech.tifr.res.in/



"What good is a faster computer, faster modem and faster printer if you're still using the same old slow fingers?"

Times of India, 19-OCT-1998