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

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 vocabulary (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 \rightarrow \text{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

\[ \nu = \frac{c}{\lambda} = \frac{34000}{4 \times 17} = 500 \text{Hz} \]

Source \rightarrow Filter \rightarrow Output

glottal vibration \hspace{1cm} vocal tract \hspace{1cm} speech wave

Formant === pole of a filter
Source-Filter model of speech production

\[ s(n) = e(n) \ast h(n) \]
\[ S(k) = E(k)H(k) \]
\[ \log(|S(k)| \ast 2) = \log(|E(k)| \ast 2) + \log(|H(k)| \ast 2) \]
Illustration in spectral domain

source: http://www.haskins.yale.edu/haskins/HEADS/MMSP/acoustic.html
Speech Spectra of /th/ and /i/ sounds
Cepstral Analysis

\[ \text{cep}(q) = \text{IFFT}\{\log(|S(k)| \ast \ast 2)\} \quad q = 0, 1, \ldots N - 1 \]

Captures not only resonances but also anti-resonances.
**Hint from biology**

### Tonotopic Map

- **Basilar Membrane**: Changes impedance along spiral
- **Apex**: Low Frequency 20 Hz
- **Base**: High Frequency 20 kHz

### Mammalian Cochlea

- **Scala Vestibuli** (Perilymph)
- **Scala Media** (Endolymph)
- **Scala Tympani** (Perilymph)
- **Reissner’s Membrane**
- **Stria Vascularis**
- **Organ of Corti**
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 phenomenon

Non-linearities along amplitude and frequency
\[ B(m) = \sum_{k=lo(m)}^{hi(m)} |X(k)|^2 \]

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

**Mel Frequency Cepstral Coefficients**

Samudravijaya K  
TIFR, samudravijaya@gmail.com  
Introduction to Automatic Speech Recognition 17/76
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
<table>
<thead>
<tr>
<th>अ</th>
<th>आ</th>
<th>इ</th>
<th>ई</th>
<th>उ</th>
<th>ऊ</th>
<th>ए</th>
<th>ऐ</th>
<th>ऒ</th>
<th>ओ</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>A</td>
<td>i</td>
<td>I</td>
<td>u</td>
<td>U</td>
<td>e</td>
<td>E</td>
<td>o</td>
<td>O</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>क</th>
<th>ख</th>
<th>ग</th>
<th>घ</th>
<th>ङ</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td>kh</td>
<td>g</td>
<td>gh</td>
<td>ng</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>च</th>
<th>छ</th>
<th>ज</th>
<th>झ</th>
<th>ञ</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>ch</td>
<td>j</td>
<td>jh</td>
<td>nj</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ट</th>
<th>ठ</th>
<th>ड</th>
<th>ढ</th>
<th>ण</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>Th</td>
<td>D</td>
<td>Dh</td>
<td>N</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>त</th>
<th>थ</th>
<th>द</th>
<th>ध</th>
<th>न</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>th</td>
<td>d</td>
<td>dh</td>
<td>n</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>प</th>
<th>फ</th>
<th>ब</th>
<th>भ</th>
<th>म</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>ph</td>
<td>b</td>
<td>bh</td>
<td>m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>य</th>
<th>र</th>
<th>ल</th>
<th>व</th>
<th>श</th>
<th>ष</th>
<th>स</th>
<th>ह</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>r</td>
<td>l</td>
<td>w</td>
<td>sh</td>
<td>S</td>
<td>s</td>
<td>h</td>
</tr>
</tbody>
</table>

Diagram: 
- Nasal Passage
- Velic
- Uvula
- Palate Velum
- Front Back
- Alveolae
- Apex
- Root
- Pharynx
- Epiglottis
- Level of Cords
- Larynx
Speech: a dynamic signal

**Formant**: frequency of resonance: F1, F2, F3, ...
Slope and curvature of trajectory
**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

\[
\Delta Cep(n, l) = \frac{\sum_{i=-L}^{L} l \ Cep(n, l)}{\sum_{i=-L}^{L} l^2}
\]

**Delta-delta (acceleration) coefficients**

\[
\Delta^2 Cep(n, l) = \frac{\sum_{i=-L}^{L} l \ \Delta Cep(n, l)}{\sum_{i=-L}^{L} l^2}
\]

Speech signal \( \Rightarrow \) Sequence of feature vectors
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

Input

Model Generation

Pattern Matching

Output

Training

Testing

Signal Processing $\Rightarrow$ Sequence of feature vectors

Pattern Recognition

Illustration: Vowel recognition with the first 2 Formant frequencies as features
Formant space of vowels
Classification criterion

* **Euclidean Distance**

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

* **Weighted Euclidean distance**

\[ d_k^k = \sqrt{\left( \frac{x - \mu_k^k}{\sigma_k^k} \right)^2} \]
Classification criterion

* Euclidean Distance
\[ x \in C_k \text{ if } (x - \mu_k)^2 \leq (x - \mu_j)^2 \forall j \]

* Weighted Euclidean distance
\[ d^k = \sqrt{\left( \frac{x - \mu^k}{\sigma^k} \right)^2} \]

* Extension to multiple features
\[ d^k = \sqrt{\sum_i \left( \frac{x_i - \mu^k_i}{\sigma^k_i} \right)^2} \]
\[ d(\bar{x}, \bar{\mu}_k) \]
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:

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

Refer to vowel F1-F2 diagram
Gaussian Mixture Model (GMM)

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

**Maximum Likelihood classification criterion for GMM case:**

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

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
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

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”.

Test feature vector sequence
**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

\[ d(n, m) : \text{the local distance between the } n^{th} \text{ test frame and } m^{th} \text{ reference frame.} \]

\[ D(n, m) : \text{the accumulated distance of the optimal path starting from the grid point } (1, 1) \text{ and ending at the grid point } (n, m). \]
Dynamic Time Warping

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

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

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

* 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.
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

<table>
<thead>
<tr>
<th></th>
<th>p</th>
<th>p</th>
<th>a</th>
<th>t</th>
<th>t</th>
<th>e</th>
<th>r</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference (correct spelling)</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Test sequence (just typed in text)

\[
d(v, c) = 2
\]
\[
d(v1, v2) = 1
\]
\[
d(c1, c2) = 1
\]
\[ d(v, c) = 2 \]
\[ d(v_1, v_2) = 1 \]
\[ d(c_1, c_2) = 1 \]

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

<table>
<thead>
<tr>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

Test sequence (just typed in text)
\[ d(v, c) = 2 \]
\[ d(v_1, v_2) = 1 \]
\[ d(c_1, c_2) = 1 \]

\[ D(x, y) = d(x, y) \]
\[ + \min \left\{ D(x-1, y-1), D(x-1, y), D(x, y-1) \right\} \]
<table>
<thead>
<tr>
<th></th>
<th></th>
<th>p</th>
<th>a</th>
<th>t</th>
<th>e</th>
<th>r</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

\[
D(x, y) = d(x, y) + \min \left\{ \begin{array}{l}
D(x-1, y-1) \\
D(x-1, y) \\
D(x, y-1)
\end{array} \right.
\]

\[
d(v, c) = 2 \\
d(v_1, v_2) = 1 \\
d(c_1, c_2) = 1
\]

Training: Viterbi (forced) alignment to get phoneme boundaries
Reference template generation: average frames belonging same phoneme
Recognition: Viterbi traceback to retrieve phoneme sequence
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
No well defined boundaries between linguistic units
Fig. 6. Diversity of transducer characteristics in telephone set [25].
Spectrogram of thiruvananthapuram
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

\[ D(n, m) = d(n, m) + \min \left\{ \begin{array}{l} D(n-1, m) \\ D(n-1, m-1) \\ D(n, m-1) \end{array} \right\} \]

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

\[ \psi_j(t) = \log(b_j(o_t)) + \max_i \{\psi_i(t-1) + \log(a_{ij})\} \]

Initial conditions: \( \psi_1(1) = 0; \psi_j(1) = \log(a_{1j}) + \log(b_j(o_1)) \)
The HMM can represent even a sentence!

source: The HTK Book
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

\[\text{===> Sentence hypothesis}\]

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
Let $Y$: Acoustic feature sequence
$W$: Word sequence

$$\hat{W} = \arg\max_{W} P(W|Y)$$
Combining Acoustic and Language Models

Let $Y$: Acoustic feature sequence
$W$: Word sequence

$\hat{W} = \arg \max_W P(W|Y)$

Bayes’ rule:

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

$\hat{W} = \arg \max_W \frac{P(Y|W)P(W)}{P(Y)}$

CSR: Acoustic model, Language model and Hypothesis search
\[ \hat{W} = \underset{\mathbf{W}}{\text{argmax}} \frac{P(Y|\mathbf{W})P(\mathbf{W})}{P(Y)} \]

Source: “State of the Art in ASR (and beyond)”, Steve Young
Basic units of HMM (phone-like units)

<table>
<thead>
<tr>
<th>अ</th>
<th>आ</th>
<th>इ</th>
<th>ई</th>
<th>उ</th>
<th>ऊ</th>
<th>ए</th>
<th>ऐ</th>
<th>ऒ</th>
<th>ओ</th>
<th>औ</th>
<th>औ</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>A</td>
<td>i</td>
<td>I</td>
<td>u</td>
<td>U</td>
<td>e</td>
<td>E</td>
<td>o</td>
<td>O</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>क</th>
<th>ख</th>
<th>ग</th>
<th>घ</th>
<th>ङ</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td>kh</td>
<td>g</td>
<td>gh</td>
<td>ng</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>च</th>
<th>छ</th>
<th>ज</th>
<th>झ</th>
<th>ञ</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>ch</td>
<td>j</td>
<td>jh</td>
<td>nj</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ट</th>
<th>ठ</th>
<th>ड</th>
<th>ढ</th>
<th>ण</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>Th</td>
<td>D</td>
<td>Dh</td>
<td>N</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>त</th>
<th>थ</th>
<th>द</th>
<th>ध</th>
<th>न</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>th</td>
<td>d</td>
<td>dh</td>
<td>n</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>प</th>
<th>फ</th>
<th>ब</th>
<th>भ</th>
<th>म</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>ph</td>
<td>b</td>
<td>bh</td>
<td>m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>य</th>
<th>र</th>
<th>ल</th>
<th>व</th>
<th>श</th>
<th>ष</th>
<th>स</th>
<th>ह</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>r</td>
<td>l</td>
<td>w</td>
<td>sh</td>
<td>s</td>
<td>h</td>
<td></td>
</tr>
</tbody>
</table>
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:

\[ c \text{ ae n t} \rightarrow c \text{ ae t} \text{ where ae is nasalised} \]

Assimilation causing changes in place of articulation:

\[ n \rightarrow m \text{ before labial stop as in input, can be, grampa} \]

Asynchronous articulation errors causing stop insertions:

\[ \text{warm[p]th, ten[t]th, on[t]ce, leng[k]th} \]

Fast speech:

\[ \text{probably} \rightarrow \text{probly} \]

r-insertion in vowel-vowel transitions:

\[ \text{stir [r]up, director [r]of} \]

Context dependent deletion:

\[ \text{nex[t] week} \]

Source: “State of the Art in ASR (and beyond)”, Steve Young
Dialect and Accent (native/non-native speakers)

* seek a dynamic speaker specific pron dictionary.
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.
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.
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
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 $\rightarrow$ Word lattice $\rightarrow$ 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

$p(\text{sil})$

sil ah

w n

ax oh

one
Lattice of phone hypotheses $\rightarrow$ 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.

Integration of syntax, semantics and domain knowledge
Statistical model: n-grams

Probability of a word sequence

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

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

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

unigram:

\[ p(w) = \frac{C(w)}{|V|} \]
Parameter Estimation

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

unigram:

\[ p(w) = \frac{C(w)}{|V|} \]

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:

\[ p(w) = \frac{C(w)}{|V|} \]

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:

\[ p(w_n|w_1w_2 \cdots w_{n-1}) = \frac{C(w_1, w_2, \cdots, w_{n-1}, w_n)}{C(w_1, w_2, \cdots, w_{n-1})} \]
Data sparsity \(\Rightarrow\) 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 occurring 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 occurred $c$ times in the corpus.
For bigrams that never occurred, the revised count is

$$c^* = \frac{N_1}{N_0}$$
Let $N_c$ denote the number of bigrams that occurred $c$ times in the corpus.
For bigrams that never occurred, the revised count is

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

In general,

$$c^* = (c + 1)\frac{N_{c+1}}{N_c}$$
Let $N_c$ denote the number of bigrams that occurred $c$ times in the corpus.

For bigrams that never occurred, 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
Linear interpolation of n-grams

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

with \( \lambda_i > 0; \quad \sum_i \lambda_i = 1.0 \)
Using n-gram ‘hierarchy’: Combining frequencies

Linear interpolation of n-grams

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

with \( \lambda_i > 0; \sum_i \lambda_i = 1.0 \)

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.
The backoff trigram grammar is computed as

```c
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


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