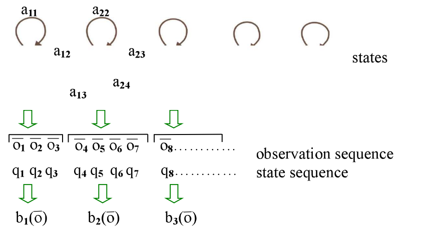
2.0 Fundamentals of Speech Recognition

References for 2.0

1.3, 3.3, 3.4, 4.2, 4.3, 6.4, 7.2, 7.3, of Bechetti

Hidden Markov Models (HMM)



- **Formulation**

$\bar{o}_t=[x_1, x_2, \ldots x_D]^T$ feature vectors for a frame at time t

$q_t \in \{1,2,3\ldots N\}$ state number for feature vector $\bar{o}_t$

$A=[a_{ij}]$, $a_{ij} = \text{Prob}[\,q_t = j \mid q_{t-1} = i\,]$ state transition probability

$B=[b_j(\bar{o}),\ j=1,2,\ldots N]$ observation (emission) probability

$b_j(\bar{o}) = \sum_{k=1}^{M} c_{jk} b_{jk}(\bar{o})$ Gaussian Mixture Model (GMM)

$b_{jk}(\bar{o})$: multi-variate Gaussian distribution for the k-th mixture (Gaussian) of the j-th state

M : total number of mixtures

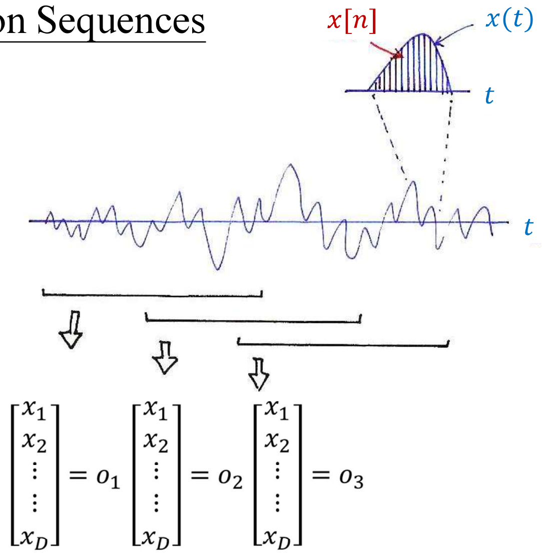
$\sum_{k=1}^{M} c_{jk} = 1$

$\pi = [\,\pi_1, \pi_2, \ldots \pi_N\,]$ initial probabilities

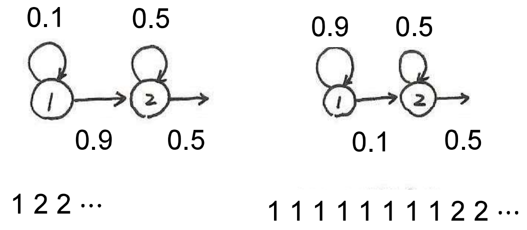
$\pi_i = \text{Prob}[q_1 = i]$

HMM : $(A, B, \pi) = \lambda$

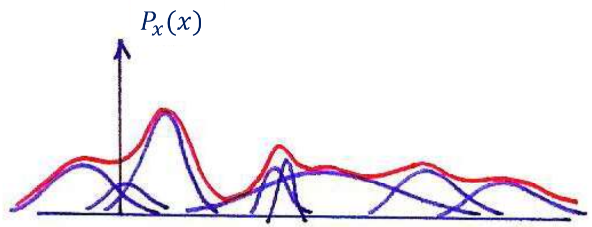
Observation Sequences



State Transition Probabilities



1-dim Gaussian Mixtures



- **Gaussian Random Variable X**

$$f_X(x) = \frac{1}{(2\pi\sigma^2)^{1/2}} e^{-(x-m)^2/2\sigma^2}$$

- **Multivariate Gaussian Distribution for n Random Variables**

$$\overline{X} = [\, X_1, X_2, \ldots\ldots, X_n \,]^t$$

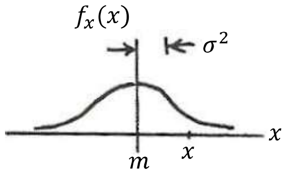
$$f_{\overline{X}}(\overline{x}) = \frac{1}{(2\pi)^{n/2}\Delta^{1/2}} e^{-\frac{1}{2}[\,(\overline{x}-\overline{\mu})^t \Sigma^{-1} (\overline{x}-\overline{\mu})\,]}$$

$$\overline{\mu} = [\, \mu_{X_1}, \mu_{X_2}, \ldots\ldots, \mu_{X_n} \,]^t$$

$\Sigma = [\, \sigma_{ij} \,]$, covariance matrix

$$\sigma_{ij} = E\,[\,(X_i - \mu_{X_i})(X_j - \mu_{X_j})\,]$$

Δ : determinant of Σ



$$\sum = \begin{array}{cc} \begin{array}{ccc} & j & \end{array} & \\ \begin{bmatrix} & \vdots & \\ \cdots & \sigma_{ij} & \cdots \\ & \vdots & \end{bmatrix} & i \end{array}$$

$$\sigma_{ij} = E[(x_i - \mu_{x_i})\,(x_j - \mu_{x_j})]$$

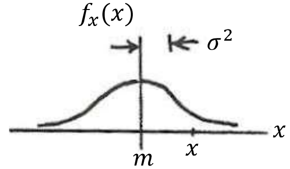
Multivariate Gaussian Distribution

$$(\bar{x}-\bar{\mu})^t \Sigma^{-1} (\bar{x}-\bar{\mu}) = \left(\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} - \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix} \right)^t \Sigma^{-1} (\bar{x}-\bar{\mu})$$

$$= [\, x_1-\mu_1 \;\; x_2-\mu_2 \;\; \cdots \;\; x_n-\mu_n \,]\, \Sigma^{-1} \begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \\ \vdots \\ x_n-\mu_n \end{bmatrix}$$

$$= (x_1-\mu_1)^2 + (x_2-\mu_2)^2 + \cdots\cdots, \quad \text{if } \Sigma = \begin{bmatrix} 1 & & 0 \\ & 1 & \\ 0 & & \ddots \end{bmatrix}$$

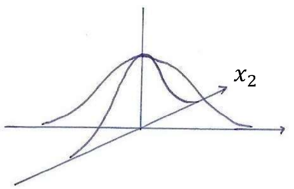
$$= \frac{(x_1-\mu_1)^2}{\sigma_{11}^2} + \frac{(x_2-\mu_2)^2}{\sigma_{22}^2} + \cdots, \quad \text{if } \Sigma = \begin{bmatrix} \sigma_{11}^2 & & 0 \\ & \sigma_{22}^2 & \\ 0 & & \ddots \; \sigma_{nn}^2 \end{bmatrix}$$

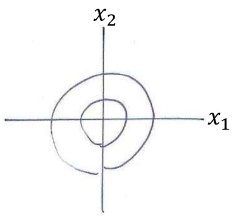


$$\sum = \begin{array}{cc} \begin{array}{ccc} & j & \end{array} & \\ \begin{bmatrix} & \vdots & \\ \cdots & \sigma_{ij} & \cdots \\ & \vdots & \end{bmatrix} & i \end{array}$$

$$\sigma_{ij} = E[(x_i - \mu_{x_i})\,(x_j - \mu_{x_j})]$$

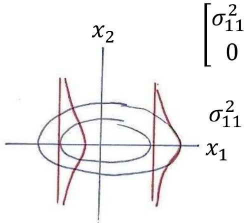
2-dim Gaussian





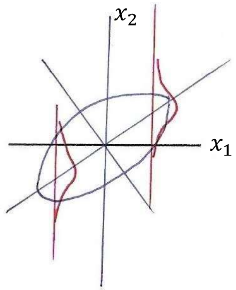
$$\begin{bmatrix} \sigma_{11}^2 & 0 \\ 0 & \sigma_{22}^2 \end{bmatrix}$$

$$\sigma_{11}^2 = \sigma_{22}^2$$



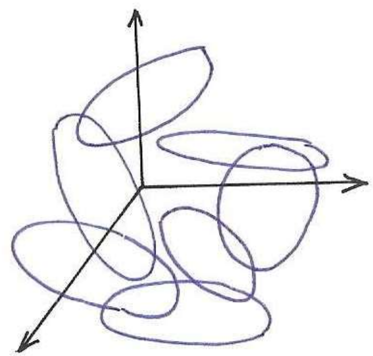
$$\begin{bmatrix} \sigma_{11}^2 & 0 \\ 0 & \sigma_{22}^2 \end{bmatrix}$$

$$\sigma_{11}^2 \neq \sigma_{22}^2$$



$$\begin{bmatrix} \sigma_{11}^2 & \sigma_{12} \\ \sigma_{21} & \sigma_{22}^2 \end{bmatrix}$$

N-dim Gaussian Mixtures



Hidden Markov Models (HMM)

- **Double Layers of Stochastic Processes**
  - hidden states with random transitions for time warping
  - random output given state for random acoustic characteristics
- **Three Basic Problems**

  (1) Evaluation Problem:

  Given $\overline{O} = (\overline{o}_1, \overline{o}_2, \ldots \overline{o}_t \ldots \overline{o}_T)$ and $\lambda = (A, B, \pi)$

  find Prob [$\overline{O} \mid \lambda$]

  (2) Decoding Problem:

  Given $\overline{O} = (\overline{o}_1, \overline{o}_2, \ldots \overline{o}_t \ldots \overline{o}_T)$ and $\lambda = (A, B, \pi)$

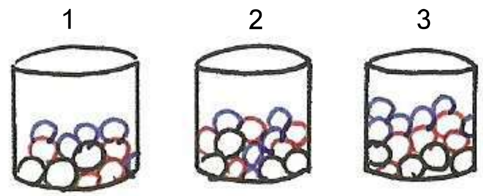
  find a best state sequence $\overline{q} = (q_1, q_2, \ldots q_t, \ldots q_T)$

  (3) Learning Problem:

  Given $\overline{O}$, find best values for parameters in λ

  such that Prob [$\overline{O} \mid \lambda$] = max

Simplified HMM



RGBGGBBGRRR……

Feature Extraction (Front-end Signal Processing)

- **Pre-emphasis**

  $H(z) = 1 - az^{-1}$, $0 << a < 1$

  $x[n] = x'[n] - ax'[n-1]$
  - pre-emphasis of spectrum at higher frequencies
- **Endpoint Detection (Speech/Silence Discrimination)**
  - short-time energy

    $$E_n = \sum_{m=-\infty}^{\infty} (x[m])^2 w[m-n]$$
  - adaptive thresholds
- **Windowing**

  $$Q_n = \sum_{m=-\infty}^{\infty} T\{\, x[m]\} w[m-n]$$

  $T\{\bullet\}$:   some operator

  $w[m]$:   window shape
  - Rectangular window

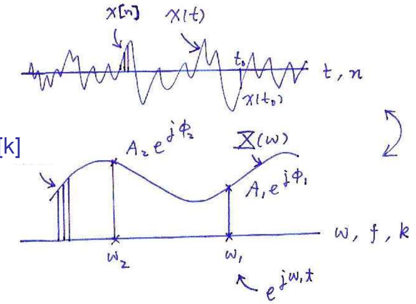
    $$w[m] = \begin{cases} 1, & 0 < m \le L-1 \\ 0, & \text{else} \end{cases}$$

    Hamming window

    $$w[m] = \begin{cases} 0.54 - 0.46\cos[\frac{2\pi m}{L}], & 0 \le m \le L-1 \\ 0, & \text{else} \end{cases}$$

    window length/shift/shape

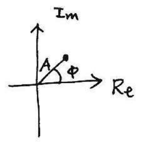
Time and Frequency Domains



time domain

1-1 mapping
Fourier Transform
Fast Fourier Transform (FFT)

frequency domain



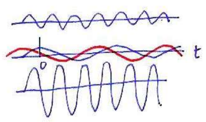
$$Re\{e^{j\omega_1 t}\} = \cos(\omega_1 t)$$

$$Re\{(A_1\, e^{j\phi_1})(e^{j\omega_1 t})\} = A_1 \cos(\omega_1 t + \phi_1)$$

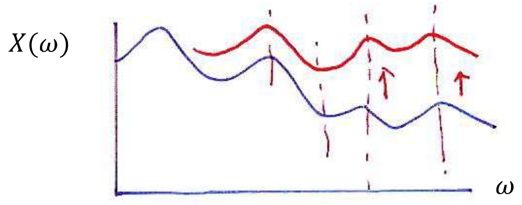
$$\vec{X} = a_1\vec{i} + a_2\vec{j} + a_3\vec{k}$$

$$\vec{X} = \sum_i a_i\, x_i$$

$$x(t) = \sum_i a_i \; x_i\,(t)$$



Pre-emphasis



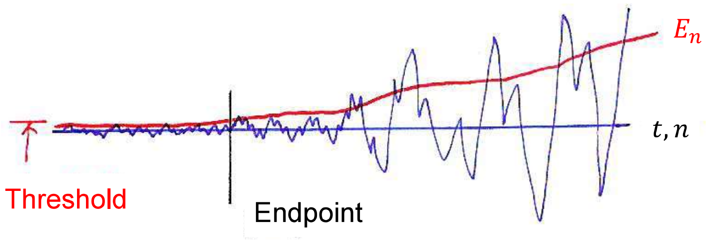
- **Pre-emphasis**

H(z) = 1 – az^{-1} ,   0 << a < 1

x[n] = x′[n] – ax′[n–1]

pre-emphasis of spectrum at higher frequencies

Endpoint Detection



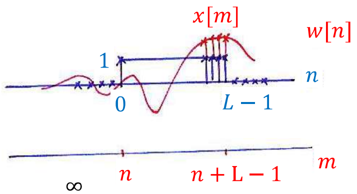
- **Endpoint Detection (Speech/Silence Discrimination)**
  - short-time energy

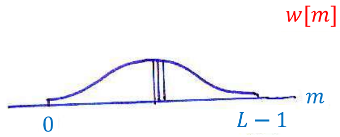
$$E_n = \sum_{m=-\infty}^{\infty} (x[m])^2\, w[m-n]$$

  - adaptive thresholds

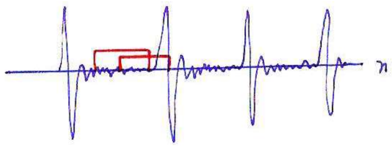
Endpoint Detection

⊙ Rectangular Window   ⊙ Hamming Window





$$E_n = \sum_{-\infty}^{\infty} (x[m])^2\, w[m-n]$$



Hamming window

$$w[m] = \begin{cases} 0.54 - 0.46\cos[\frac{2\pi m}{L}], 0 \le m \le L-1 \\ 0, \text{ else} \end{cases}$$

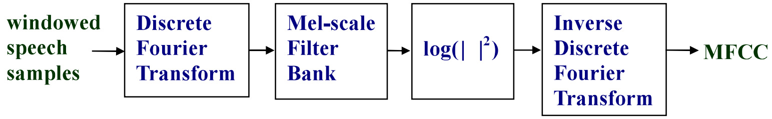
$$Q_n = \sum_{m=-\infty}^{\infty} T\{x[m]\}\, w[m-n]$$

$T\{\bullet\}$: some operator

w$[m]$: window shape

Feature Extraction (Front-end Signal Processing)

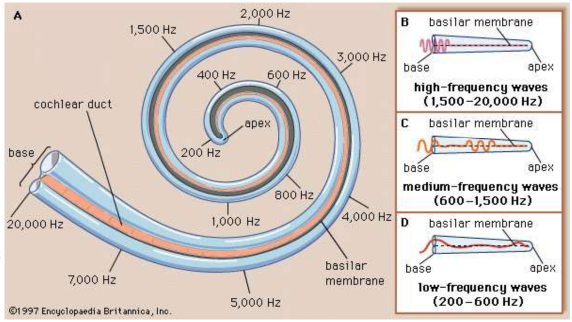
- **Mel Frequency Cepstral Coefficients (MFCC)**



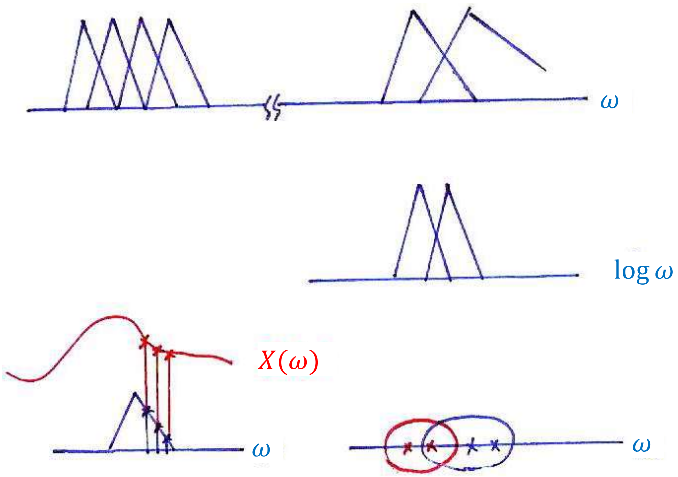
- Mel-scale Filter Bank
  triangular shape in frequency/overlapped
  uniformly spaced below 1 kHz
  logarithmic scale above 1 kHz

- **Delta Coefficients**
  - 1st/2nd order differences

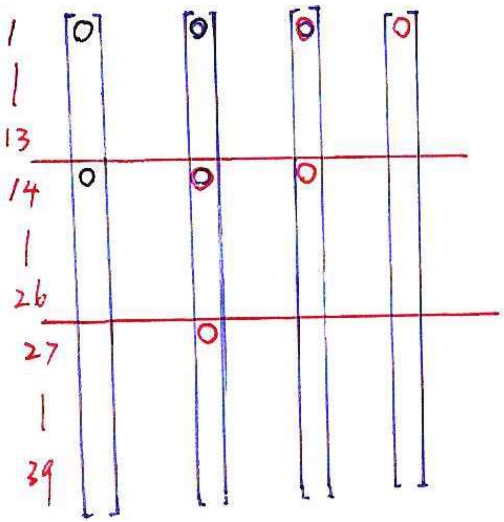
Peripheral Processing for Human Perception (P.34 of 7.0)



Mel-scale Filter Bank



Delta Coefficients



Language Modeling: N-gram

$W = (w_1, w_2, w_3, \ldots, w_i, \ldots w_R)$ a word sequence

- Evaluation of P(W)

$P(W) = P(w_1) \prod_{i=2}^{R} P(w_i|w_1, w_2, \ldots w_{i-1})$

- Assumption:

$P(w_i|w_1, w_2, \ldots w_{i-1}) = P(w_i|w_{i-N+1}, w_{i-N+2}, \ldots w_{i-1})$

Occurrence of a word depends on previous N−1 words only

N-gram language models

N = 2	:	bigram	$P(w_i \mid w_{i-1})$
N = 3	:	tri-gram	$P(w_i \mid w_{i-2}, w_{i-1})$
N = 4	:	four-gram	$P(w_i \mid w_{i-3}, w_{i-2}, w_{i-1})$
⋮			
N = 1	:	unigram	$P(w_i)$

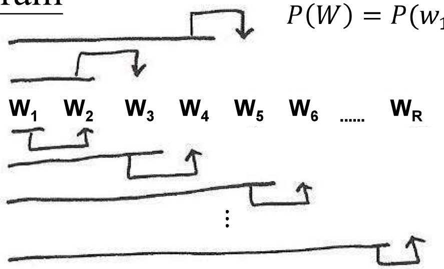
probabilities estimated from a training text database

example : tri-gram model

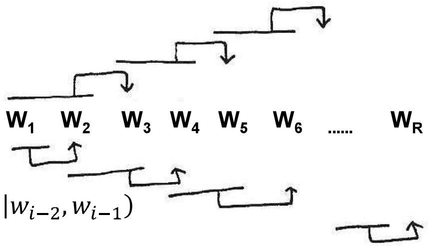
$P(W) = P(w_1)\, P(w_2|w_1) \prod_{i=3}^{N} P(w_i|w_{i-2}, w_{i-1})$

N-gram

$$P(W) = P(w_1)\prod_{i=2}^{R} P\left(w_i | w_1, w_2, \cdots, w_{i-1}\right)$$



⊙ tri-gram



$$P(W) = P(w_1)P(w_2|w_1)\prod_{i=3}^{N} P\left(w_i | w_{i-2}, w_{i-1}\right)$$

Language Modeling

- Evaluation of N-gram model parameters

unigram

$$P(w^i) = \frac{N(w^i)}{\sum_{j=1}^{V} N(w^j)}$$

w^i : a word in the vocabulary

V : total number of different words in the vocabulary

N(•) number of counts in the training text database

bigram

$$P(w^j|w^k) = \frac{N(<w^k, w^j>)}{N(w^k)}$$

$< w^k, w^j >$: a word pair

trigram

$$P(w^j|w^k, w^m) = \frac{N(<w^k, w^m, w^j>)}{N(<w^k, w^m>)}$$

smoothing – estimation of probabilities of rare events by statistical approaches

… this ………	50000
… this is ……	500
… this is a …	5

$$\text{Prob [is| this]} = \frac{500}{50000}$$

$$\text{Prob [a| this is]} = \frac{5}{500}$$

bigram

$$P\left(w^j | w^k\right) = \frac{N(\langle w^k, w^j\rangle)}{N(w^k)}$$

$\langle w^k, w^j \rangle$: a word pair

trigram

$$P\left(w^j | w^k, w^m\right) = \frac{N(\langle w^k, w^m, w^j\rangle)}{N(\langle w^k, w^m\rangle)}$$

Large Vocabulary Continuous Speech Recognition

$\overline{W} = (w_1, w_2, \ldots w_R)$ a word sequence

$\overline{O} = (\bar{o}_1, \bar{o}_2, \ldots \bar{o}_T)$ feature vectors for a speech utterance

$W^* = \underset{W}{\text{Arg Max}}\ \text{Prob}(W|\overline{O})$ MAP principle

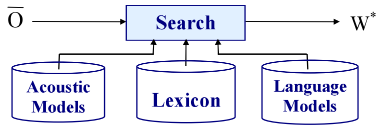
$\text{Prob}(W|\overline{O}) = \frac{\text{Prob}(\overline{O}|W) \bullet P(W)}{P(\overline{O})} = \max$ A Posteriori Probability

Maximum A Posteriori (MAP) Principle

$\text{Prob}(\overline{O}|W) \bullet P(W) = \max$

by HMM by language model

- **A Search Process Based on Three Knowledge Sources**



- Acoustic Models : HMMs for basic voice units (e.g. phonemes)
- Lexicon : a database of all possible words in the vocabulary, each word including its pronunciation in terms of component basic voice units
- Language Models : based on words in the lexicon

Maximum A Posteriori Principle (MAP)

W : { w_1 , w_2 , w_3 }

↑ sunny ↑ rainy ↑ cloudy

$P(w_1)$
$P(w_2)$
$+\ P(w_3)$

1.0

$\vec{O} = (\vec{o}_1, \vec{o}_2, \vec{o}_3, \cdots)$ weather parameters

⊙ Problem

given $\vec{O}$ today, to predict W for tomorrow

Maximum A Posteriori Principle (MAP)

⊙ Approach 1

Comparing $P(w_1)$, $P(w_2)$, $P(w_3)$

$\vec{O}$ not used?

⊙ Approach 2

A Posteriori Probability 事後機率

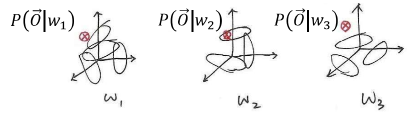
Likelihood function

Prior Probability 事前機率

compute $P\left(\begin{matrix} w_1 \\ w_2 \\ w_3 \end{matrix} \middle| \vec{O}\right) = \dfrac{P\left(\vec{O} \middle| \begin{matrix} w_1 \\ w_2 \\ w_3 \end{matrix}\right) \cdot P\left(\begin{matrix} w_1 \\ w_2 \\ w_3 \end{matrix}\right)}{P(\vec{O})}$, $P\left(w_i \middle| \vec{O}\right) = \dfrac{P(\vec{O}|w_i)\,P(w_i)}{P(\vec{O})}$, $i = 1, 2, 3$

unknown observation

compare $P\left(\vec{O} \middle| \begin{matrix} w_1 \\ w_2 \\ w_3 \end{matrix}\right) \cdot P\left(\begin{matrix} w_1 \\ w_2 \\ w_3 \end{matrix}\right)$, $P(\vec{O}|w_i) \cdot P(w_i)$, $i = 1, 2, 3$



Syllable-based One-pass Search

- **Finding the Optimal Sentence from an Unknown Utterance Using 3 Knowledge Sources: Acoustic Models, Lexicon and Language Model**
- **Based on a Lattice of Syllable Candidates**