제 **1**회 컴퓨터비전 및 패턴인식 겨울학교**, 2006.2.1~3, KAIST**

A Tutorial on Hidden Markov Models

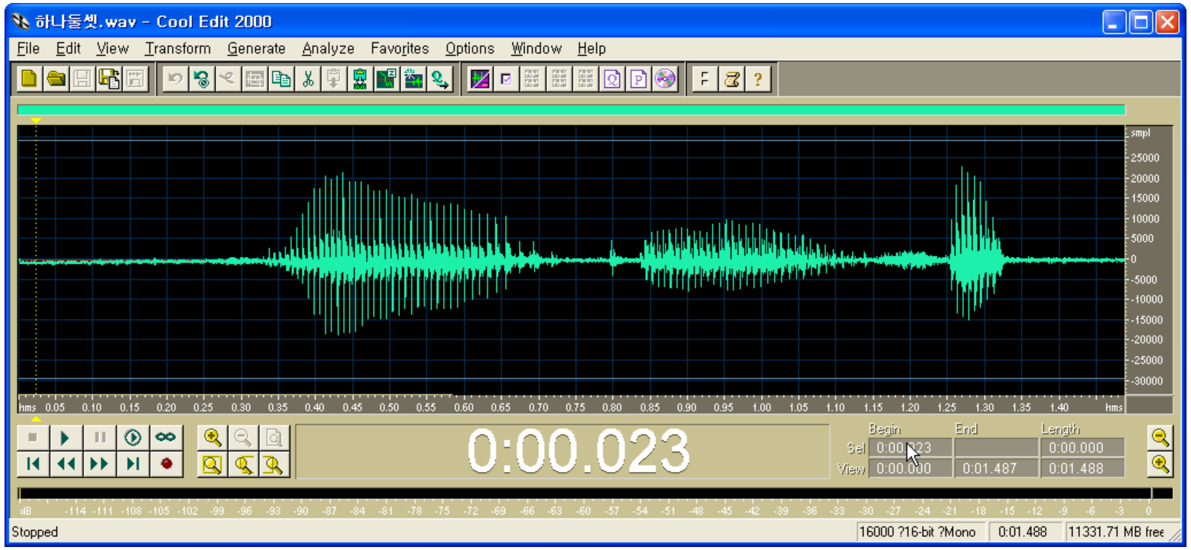
2006년 2월 2일

하진영
jyha@kangwon.ac.kr
강원대학교 컴퓨터학부

Contents

Sequential Data

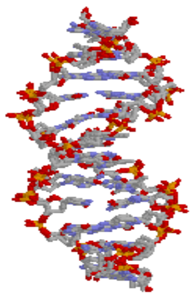
- Examples



Speech data
("하나 둘 셋")

Handwriting data

DNA



AAAAGAAAAGGTTAGAAAGATGAGAGATGATAAAGGGTCCATTTG
AGGTTAGGTAATATGGTTTGGTATCCCTGTAGTTAAAAGTTTTTG
TCTTATTTTAGAATACTGTGACTATTTCTTTAGTATTAATTTTTC
CTTCTGTTTTCCTCATCTAGGGAACCCCAAGAGCATCCAATAGAA
GCTGTGCAATTATGTAAAATTTTCAACTGTCTTCCTCAAAATAAA
GAAGTATGGTAATCTTTACCTGTATACAGTGCAGAGCCTTCTCAG
AAGCACAGAATATTTTTATATTTCCTTTATGTGAATTTTTAAGCT
GCAAATCTGATGGCCTTAATTTCCTTTTTGACACTGAAAGTTTTG
TAAAAGAAATCATGTCCATACACTTTGTTGCAAGATGTGAATTAT
TGACACTGAACTTAATAACTGTGTACTGTTCGGAAGGGGTTCCTC
AAATTTTTTGACTTTTTTTGTATGTGTGTTTTTTCTTTTTTTTTA
AGTTCTTATGAGGAGGGAGGGTAAATAAACCACTGTGCGTCTTGG
TGTAATTTGAAGATTGCCCCATCTAGACTAGCAATCTCTTCATTA
TTCTCTGCTATATATAAAACGGTGCTGTGAGGGAGGGGAAAAGCA
TTTTTCAATATATTGAACTTTTGTACTGAATTTTTTTTGTAATAAG
CAATCAAGGTTATAATTTTTTTTAAAATAGAAATTTTGTAAGAAG
GCAATATTAACCTAATCACCATGTAAGCACTCTGGATGATGGATT
CCACAAAACTTGGTTTTATGGTTACTTCTTCTCTTAGATTCTTAA
TTCATGAGGAGGGTGGGGGAGGGAGGTGGAGGGAGGGAAGGGTTT
CTCTATTAAAATGCATTCGTTGTGTTTTTTAAGATAGTGTAACTT
GCTAAATTTCTTATGTGACATTAACAAATAAAAAAGCTCTTTTAA
TATTAGATAA

18

Characteristics of such data

- ✓ Data are sequentially generated according to time or index
- ✓ Spatial information along time or index
- ✓ Often highly variable, but has an embedded structure
- ✓ Information is contained in the structure

Advantage of HMM on Sequential Data

- Natural model structure: doubly stochastic process
  - transition parameters model *temporal* variability
  - output distribution model *spatial* variability
- Efficient and good modeling tool for
  - sequences with temporal constraints
  - spatial variability along the sequence
  - real world complex processes
- Efficient evaluation, decoding and training algorithms
  - Mathematically strong
  - Computationally efficient
- Proven technology!
  - Successful stories in many applications
- Tools already exist
  - HTK (Hidden Markov Model Toolkit)
  - HMM toolbox for Matlab

Successful Application Areas of HMM

- On-line handwriting recognition
- Speech recognition and segmentation
- Gesture recognition
- Language modeling
- Motion video analysis and tracking
- Protein sequence/gene sequence alignment
- Stock price prediction
- …

What’s HMM?

Hidden Markov Model

Hidden **+** Markov Model

What is ‘hidden’?

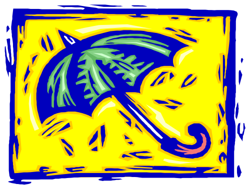
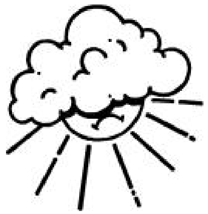
What is ‘Markov model’?

Markov Model

- Scenario
- Graphical representation
- Definition
- Sequence probability
- State probability

Markov Model: Scenario

- Classify a weather into three states
  - State 1: rain or snow
  - State 2: cloudy
  - State 3: sunny

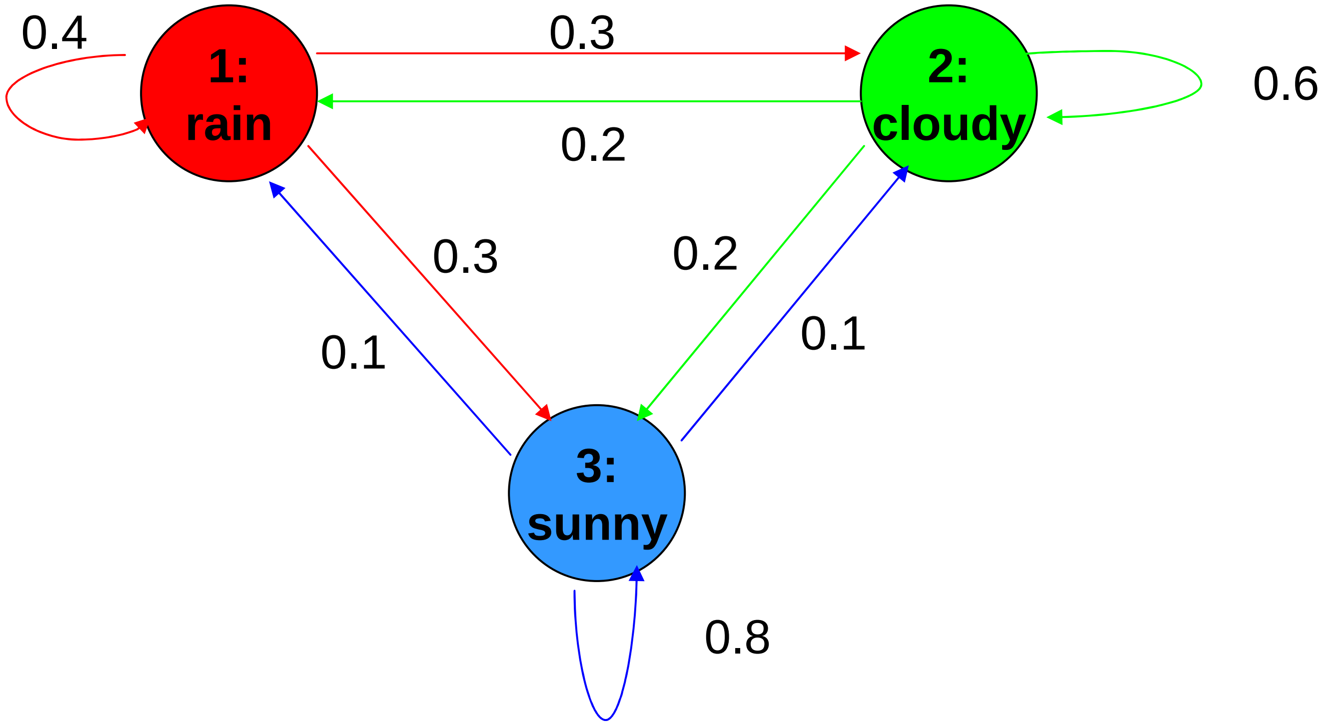

- By carefully examining the weather of some city for a long time, we found following weather change pattern

		Tomorrow		
		Rain/snow	Cloudy	Sunny
Today	Rain/Snow	0.4	0.3	0.3
	Cloudy	0.2	0.6	0.2
	Sunny	0.1	0.1	0.8

Assumption: tomorrow weather depends only on today's weather!

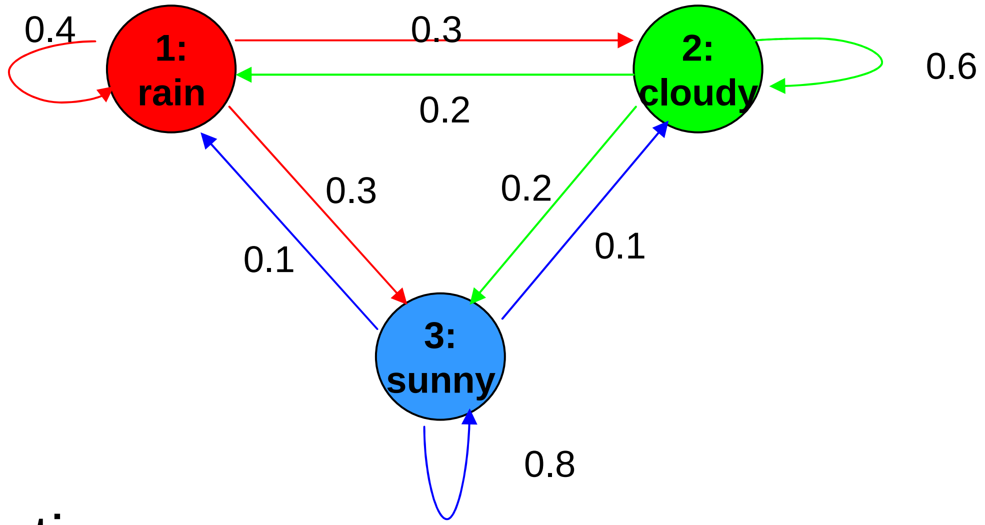
Markov Model: Graphical Representation

- Visual illustration with diagram



- Each state corresponds to one observation
- Sum of outgoing edge weights is one

Markov Model: Definition



- Observable states

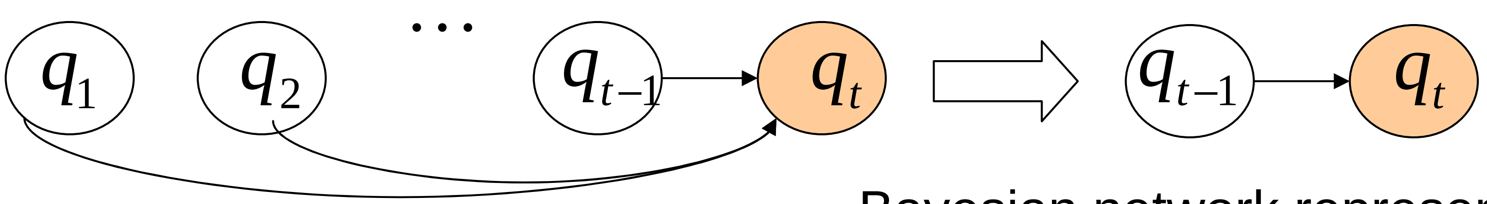
$$\{1,\ 2,\ \cdots,\ N\}$$

- Observed sequence

$$q_1, q_2, \cdots, q_T$$

- 1st order Markov assumption

$$P(q_t = j \mid q_{t-1} = i, q_{t-2} = k, \cdots) = P(q_t = j \mid q_{t-1} = i)$$



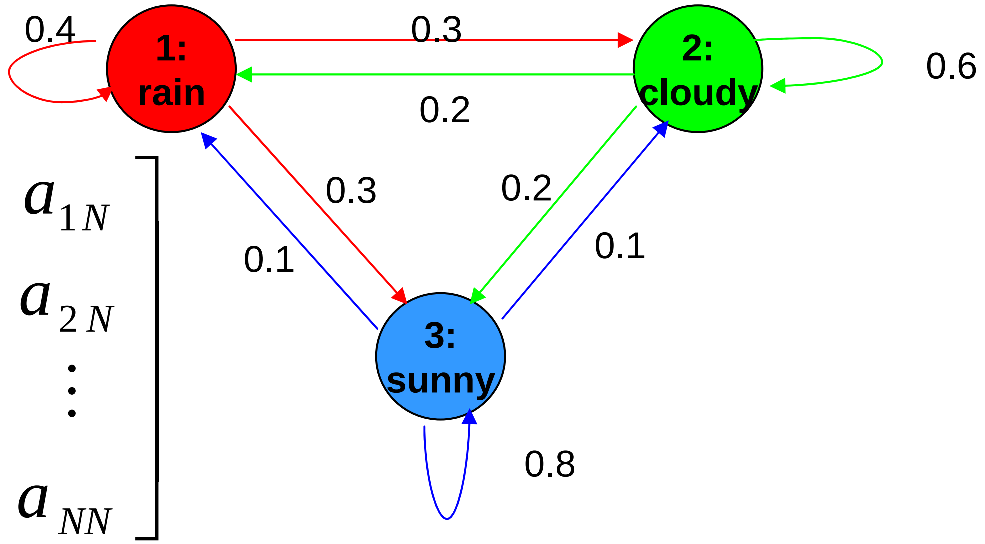
Bayesian network representation

- Stationary

$$P(q_t = j \mid q_{t-1} = i) = P(q_{t+l} = j \mid q_{t+l-1} = i)$$

Markov Model: Definition (Cont.)

- State transition matrix



$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{NN} & \cdots & a_{NN} \end{bmatrix}$$

- Where

$$a_{ij} = P(q_t = j \mid q_{t-1} = i), \qquad 1 \le i, j \le N$$

- With constraints

$$a_{ij} \ge 0, \qquad \sum_{j=1}^{N} a_{ij} = 1$$

- Initial state probability

$$\pi_i = P(q_1 = i), \qquad 1 \le i \le N$$

Markov Model: Sequence Prob.

- Conditional probability

$$P(A,B) = P(A \mid B)P(B)$$

- Sequence probability of Markov model

Chain rule

$$\begin{aligned}
&P(q_1, q_2, \cdots, q_T) \\
&= P(q_1)P(q_2 \mid q_1)\cdots P(q_{T-1} \mid q_1, \cdots, q_{T-2})P(q_T \mid q_1, \cdots, q_{T-1}) \\
&= P(q_1)P(q_2 \mid q_1)\cdots P(q_{T-1} \mid q_{T-2})P(q_T \mid q_{T-1})
\end{aligned}$$

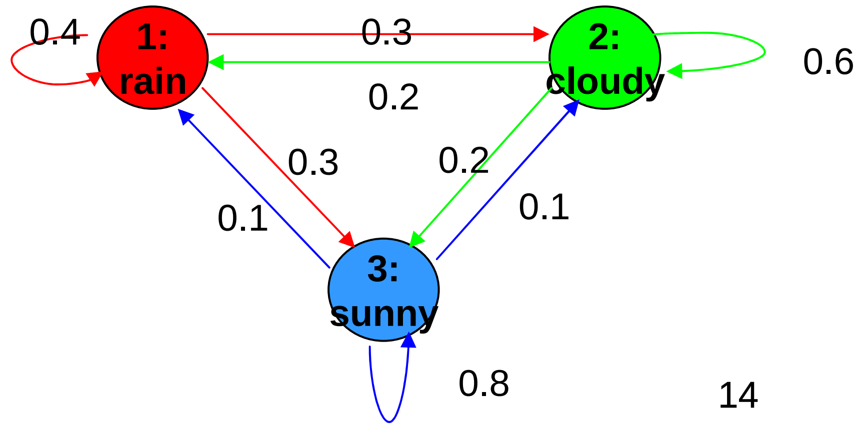
1st order Markov assumption

Markov Model: Sequence Prob. (Cont.)

- Question: What is the probability that the weather for the next 7 days will be "sun-sun-rain-rain-sun-cloudy-sun" when today is sunny?

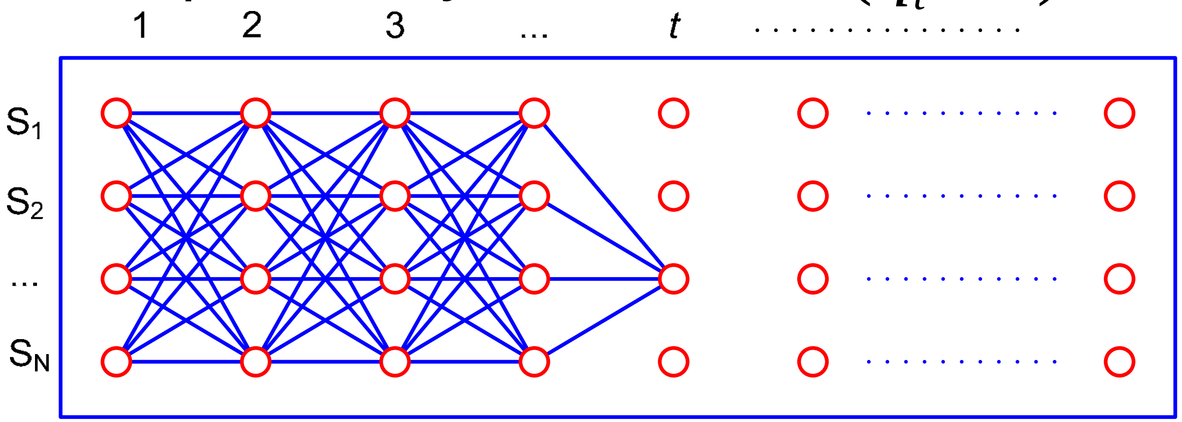
$$S_1 : rain, \quad S_2 : cloudy, \quad S_3 : sunny$$

$$\begin{aligned}
P(O \mid \text{model}) &= P(S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3 \mid \text{model}) \\
&= P(S_3) \cdot P(S_3 \mid S_3) \cdot P(S_3 \mid S_3) \cdot P(S_1 \mid S_3) \\
&\quad \cdot P(S_1 \mid S_1) P(S_3 \mid S_1) P(S_2 \mid S_3) P(S_3 \mid S_2) \\
&= \pi_3 \cdot a_{33} \cdot a_{33} \cdot a_{31} \cdot a_{11} \cdot a_{13} \cdot a_{32} \cdot a_{23} \\
&= 1 \cdot (0.8)(0.8)(0.1)(0.4)(0.3)(0.1)(0.2) \\
&= 1.536 \times 10^{-4}
\end{aligned}$$



Markov Model: State Probability

- State probability at time *t* : $P(q_t = i)$



- Simple but slow algorithm:
  – Probability of a path that ends to state *i* at time *t*:

$$Q_t(i) = (q_1, q_2, \cdots, q_t = i)$$

$$P(Q_t(i)) = \pi_{q_1} \prod_{k=2}^{t} P(q_k \mid q_{k-1})$$

  – Summation of probabilities of all the paths that ends to *i* at *t*

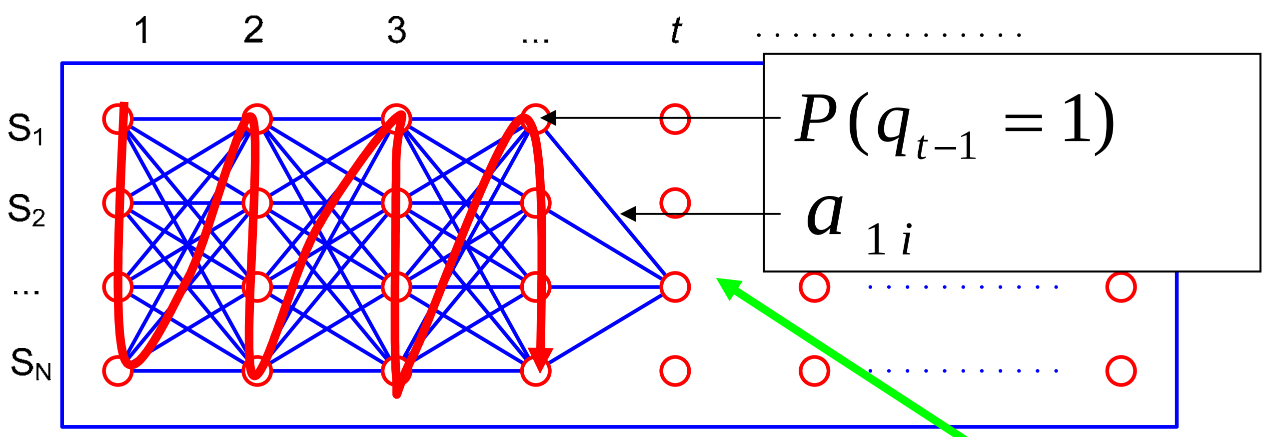
$$P(q_t = i) = \sum_{all\ \ Q_t(i)'s} P(Q_t(i))$$

Exponential time complexity:

$O(N^t)$

Markov Model: State Prob. (Cont.)

- State probability at time *t* : $P(q_t = i)$



Each node stores the sum of probabilities of partial paths

- Efficient algorithm (Lattice algorithm)
  – Recursive path probability calculation

$$P(q_t = i) = \sum_{j=1}^{N} P(q_{t-1} = j, q_t = i)$$

$$= \sum_{j=1}^{N} P(q_{t-1} = j) P(q_t = i \mid q_{t-1} = j)$$

$$= \sum_{j=1}^{N} P(q_{t-1} = j) \cdot a_{ji}$$

Time complexity: $O(N^2 t)$

What's HMM?

Hidden Markov Model

Hidden + Markov Model

What is 'hidden'?

What is 'Markov model'?

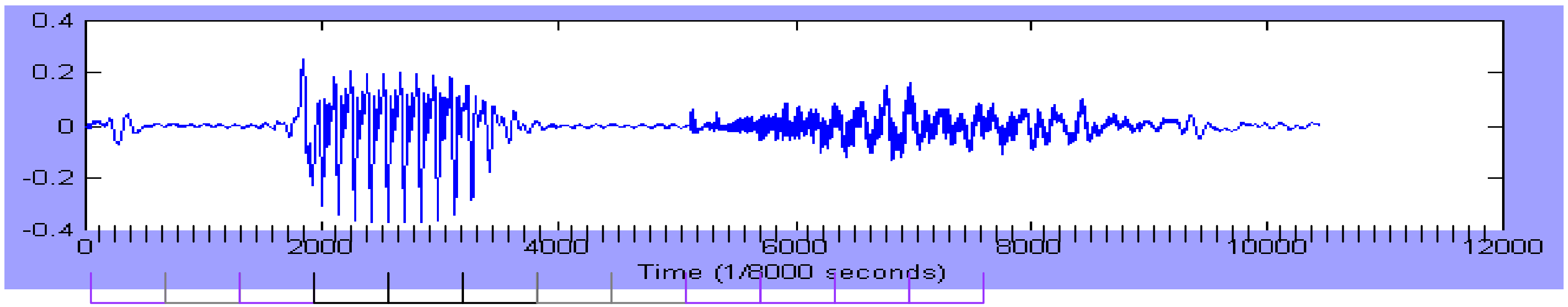
Hidden Markov Model

- Example
- Generation process
- Definition
- Model evaluation algorithm
- Path decoding algorithm
- Training algorithm

Time Series Example

- Representation
  - $\mathbf{X} = \mathbf{x_1}\ \mathbf{x_2}\ \mathbf{x_3}\ \mathbf{x_4}\ \mathbf{x_5} \ldots \mathbf{x_{T-1}}\ \mathbf{x_T}$

    = s ϕ p iy iy iy ϕ ϕ ch ch ch ch



Analysis Methods

- Probability-based analysis?

$$P(\mathrm{s}\ \phi\ \mathrm{p\ iy\ iy\ iy}\ \phi\ \phi\ \mathrm{ch\ ch\ ch\ ch}) = ?$$

- Method I

$$\boxed{P(\mathrm{s})P(\phi)^3 P(\mathrm{p})P(\mathrm{iy})^3 P(\mathrm{ch})^4}$$

  – Observations are independent; no time/order

  – A poor model for temporal structure

    - Model size = $|V| = N$

Analysis methods

- Method II

$$P(\text{s})P(\text{s}\,|\,\text{s})P(\phi\,|\,\text{s})P(\text{p}\,|\,\phi)P(\text{iy}\,|\,\text{p})P(\text{iy}\,|\,\text{iy})^2$$
$$\times P(\phi\,|\,\text{iy})P(\phi\,|\,\phi)P(\text{ch}\,|\,\phi)P(\text{ch}\,|\,\text{ch})^{2`}$$

– A simple model of ordered sequence

- A symbol is dependent only on the immediately preceding:

$$P(x_t \mid x_1 x_2 x_3 \cdots x_{t-1}) = P(x_t \mid x_{t-1})$$

- $|V| \times |V|$ matrix model
- 50×50 – not very bad …
- $10^5 \times 10^5$ – doubly outrageous!!

The problem

- "What you see is the truth"
  - Not quite a valid assumption
  - There are often errors or noise
    - Noisy sound, sloppy handwriting, ungrammatical sentences
  - There may be some truth process
    - Underlying hidden sequence
    - Obscured by the incomplete observation

Another analysis method

- Method III
  - What you see is a clue to <u>what lies behind</u> and is not known *a priori*
    - The source that generated the observation
    - The source evolves and generates characteristic observation sequences

$$q_0 \to q_1 \to q_2 \to \quad \cdots \quad \to q_T$$

$$P(\mathrm{s}, q_1) P(\mathrm{s}, q_2 \mid q_1) P(\phi, q_3 \mid q_2) \cdots P(\mathrm{ch}, q_T \mid q_{T-1}) = \prod_t P(x_t, q_t \mid q_{t-1})$$

$$\sum_Q P(\mathrm{s}, q_1) P(\mathrm{s}, q_2 \mid q_1) P(\phi, q_3 \mid q_2) \cdots P(\mathrm{ch}, q_T \mid q_{T-1}) = \sum_Q \prod_t P(x_t, q_t \mid q_{t-1})$$

The Auxiliary Variable

$$q_t \in S = \{1, \ldots, N\}$$

- N is also conjectured
- $\{q_t : t \geq 0\}$ is conjectured, not visible
  - is $Q = q_1 q_2 \cdots q_T$
  - is Markovian

$$P(q_1 q_2 \cdots q_T) = P(q_1) P(q_2 \mid q_1) \cdots P(q_T \mid q_{T-1})$$

  - “Markov chain”

Summary of the Concept

$$
\begin{aligned}
P(X) &= \sum_{Q} P(X, Q) \\
&= \sum_{Q} P(Q) P(X \mid Q) \\
&= \sum_{Q} P(q_1 q_2 \cdots q_T) P(x_1 x_2 \cdots x_T \mid q_1 q_2 \cdots q_T) \\
&= \sum_{Q} \underbrace{\prod_{t=1}^{T} P(q_t \mid q_{t-1})}_{\text{Markov chain process}} \underbrace{\prod_{t=1}^{T} p(x_t \mid q_t)}_{\text{Output process}}
\end{aligned}
$$

Hidden Markov Model

- is a doubly stochastic process
  – stochastic chain process : $\{ q(t) \}$
  – output process : $\{ f(x|q) \}$

- is also called as
  – *Hidden Markov chain*
  – *Probabilistic function of Markov chain*

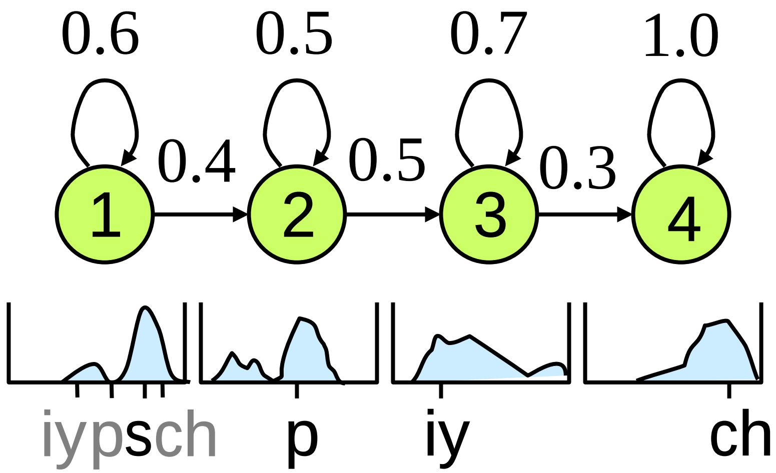
HMM Characterization

- $\lambda = (A, B, \pi)$
  - A : state transition probability
    $\{ a_{ij} \mid a_{ij} = p(q_{t+1}=j \mid q_t=i) \}$
  - B : symbol output/observation probability
    $\{ b_j(v) \mid b_j(v) = p(x=v \mid q_t=j) \}$
  - π : initial state distribution probability
    $\{ \pi_i \mid \pi_i = p(q_1=i) \}$

$$\sum_Q P(Q \mid \lambda) P(\mathbf{X} \mid Q, \lambda)$$

$$= \sum_Q \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \ldots a_{q_{T-1} q_T} b_{q_1}(x_1) b_{q_2}(x_2) \ldots b_{q_T}(x_T) \Big|_{\lambda}$$

Graphical Example



$$\pi = [\ 1.0 \quad 0 \quad 0 \quad 0\ ]$$

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0.6 & 0.4 & 0.0 & 0.0 \\ 0.0 & 0.5 & 0.5 & 0.0 \\ 0.0 & 0.0 & 0.7 & 0.3 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix} \end{array}$$

$$B = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} \text{ch} & \text{iy} & \text{p} & \text{s} & \end{array} \\ \begin{bmatrix} 0.2 & 0.2 & 0.0 & 0.6 & \ldots \\ 0.0 & 0.2 & 0.5 & 0.3 & \ldots \\ 0.0 & 0.8 & 0.1 & 0.1 & \ldots \\ 0.6 & 0.0 & 0.2 & 0.2 & \ldots \end{bmatrix} \end{array}$$

Data interpretation

$$
\begin{aligned}
&P(\text{s s p p iy iy iy ch ch ch}|\lambda) \\
&= \sum_Q P(\text{ssppiyiyiychchch}, Q|\lambda) \\
&= \sum_Q \underbrace{P(Q|\lambda)\, p(\text{ssppiyiyiychchch}|Q,\lambda)}
\end{aligned}
$$

$$
\begin{bmatrix}
\boxed{0.6} & \boxed{0.4} & 0.0 & 0.0 \\
0.0 & \boxed{0.5} & \boxed{0.5} & 0.0 \\
0.0 & 0.0 & \boxed{0.7} & \boxed{0.3} \\
0.0 & 0.0 & 0.0 & \boxed{1.0}
\end{bmatrix}
$$

$$
\begin{bmatrix}
0.2 & 0.2 & 0.0 & \boxed{0.6} & \ldots \\
0.0 & 0.2 & \boxed{0.5} & 0.3 & \ldots \\
0.0 & \boxed{0.8} & 0.1 & 0.1 & \ldots \\
\boxed{0.6} & 0.0 & 0.2 & 0.2 & \ldots
\end{bmatrix}
$$

Let Q = 1 1 2 2 3 3 3 4 4 4

$$
\begin{aligned}
&P(Q|\lambda)\, p(\text{ssppiyiyiychchch}|Q, \lambda) \\
&\quad = \mathrm{P}(1122333444|\lambda)\, p(\text{ssppiyiyiychchch}|1122333444, \lambda) \\
&\quad = (1\times.6)\times(.6\times.6)\times(.4\times.5)\times(.5\times.5)\times(.5\times.8)\times(.7\times.8)^2 \\
&\qquad\qquad \times(.3\times.6)\times(1.\times.6)^2 \\
&\quad \cong 0.0000878
\end{aligned}
$$

#multiplications ~ $2TN^T$

Issues in HMM

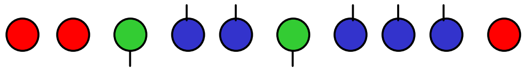
- Intuitive decisions
  1. number of states (N)
  2. topology (state inter-connection)
  3. number of observation symbols (V)

- Difficult problems
  4. efficient computation methods
  5. probability parameters (λ)

The Number of States

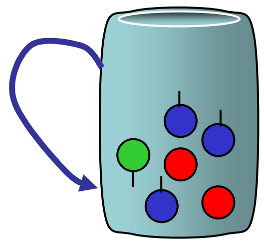
- How many states?
  - Model size
  - Model topology/structure
- Factors
  - Pattern complexity/length and variability
  - The number of samples
- Ex:

r r g b b g b b b r



(1) The simplest model

- Model I
  - $N = 1$
  - a_{11}=1.0
  - B = [1/3, 1/6, 1/2]



$$P(\mathrm{r\ r\ g\ b\ b\ g\ b\ b\ b\ r} \mid \lambda_1) = 1\times\frac{1}{3}\times1\times\frac{1}{3}\times1\times\frac{1}{6}\times1\times\frac{1}{2}\times1\times\frac{1}{2}\times1\times\frac{1}{6}$$

$$\times1\times\frac{1}{2}\times1\times\frac{1}{2}\times1\times\frac{1}{2}\times1\times\frac{1}{3}$$

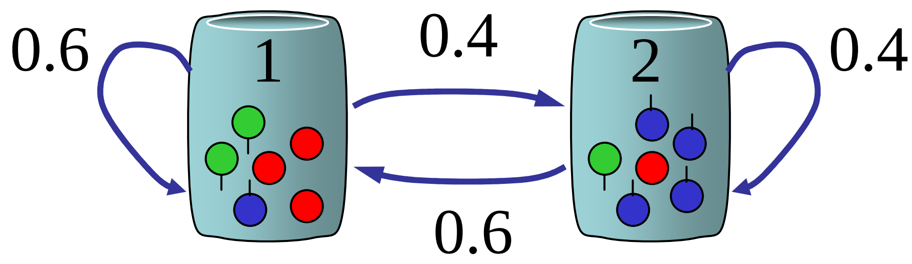
$$\cong 0.0000322 \ \ (< 0.0000338)$$

(2) Two state model

- Model II:
  - $N = 2$

$$A = \begin{bmatrix} 0.6 & 0.4 \\ 0.6 & 0.4 \end{bmatrix}$$

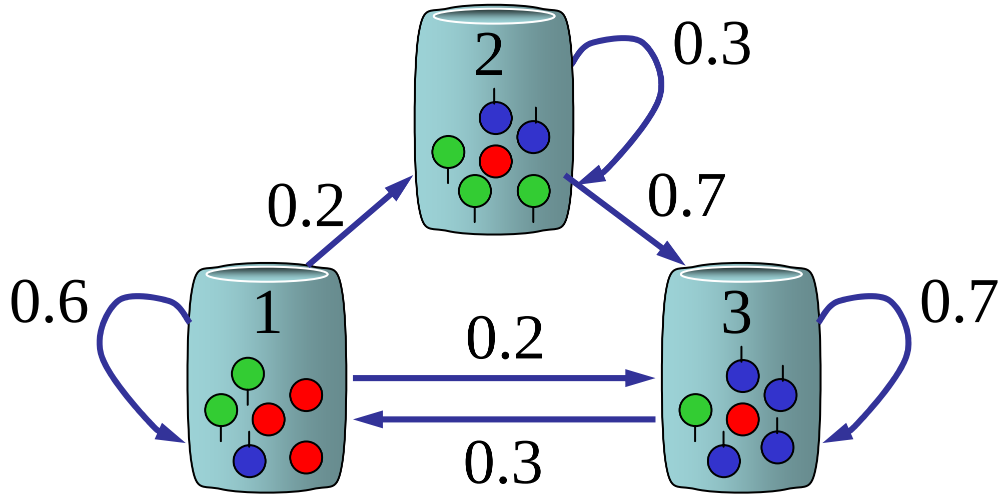
$$B = \begin{bmatrix} 1/2 & 1/3 & 1/6 \\ 1/6 & 1/6 & 2/3 \end{bmatrix}$$

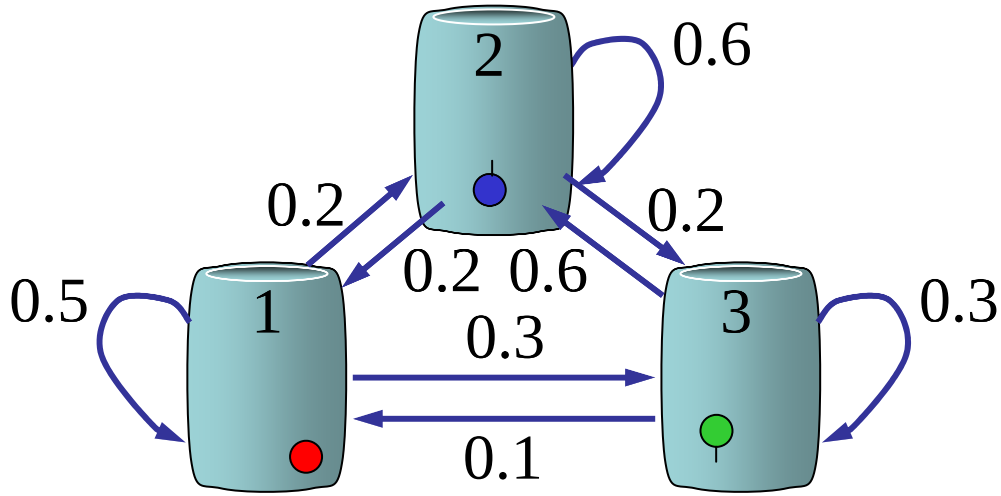


$$P(\mathrm{r\,r\,g\,b\,b\,g\,b\,b\,b\,r} \mid \lambda_1) = .5\times\frac{1}{2}\times.6\times\frac{1}{2}\times.6\times\frac{1}{3}\times.4\times\frac{2}{3}\times.4\times\frac{2}{3}\times.6\times\frac{1}{3}$$
$$\times.4\times\frac{2}{3}\times.4\times\frac{2}{3}\times.4\times\frac{2}{3}\times.6\times\frac{1}{2} \;+\; \cdots\cdots$$
$$= ?$$

(3) Three state models

- N=3:





The Criterion is

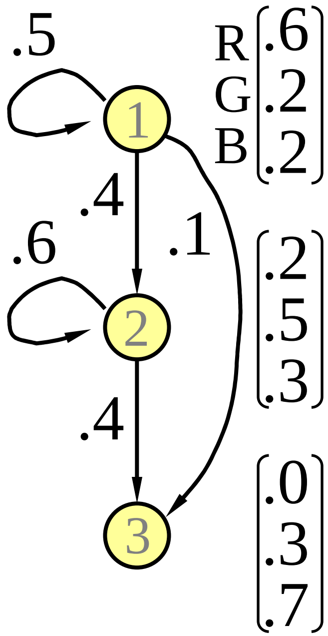
- Obtaining the best model(λ) that maximizes

$$P(X \mid \hat{\lambda})$$

- The best topology comes from insight and experience

  $\leftarrow$ the # classes/symbols/samples

A trained HMM

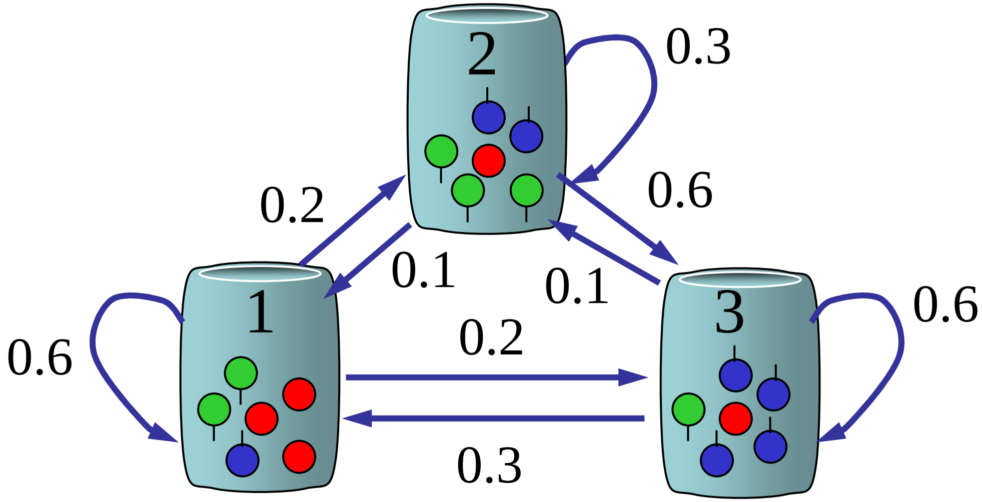


$$\pi = \begin{bmatrix} 1. & 0. & 0. \end{bmatrix}$$

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} .5 & .4 & .1 \\ .0 & .6 & .4 \\ .0 & .0 & .0 \end{pmatrix} \end{array}$$

$$B = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} R & G & B \end{array} \\ \begin{pmatrix} .6 & .2 & .2 \\ .2 & .5 & .3 \\ .0 & .3 & .7 \end{pmatrix} \end{array}$$

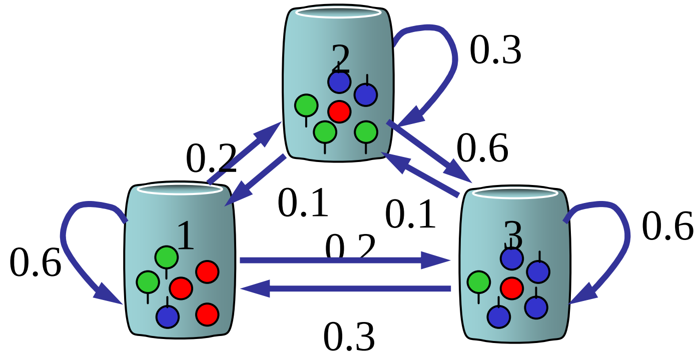
Hidden Markov Model: Example

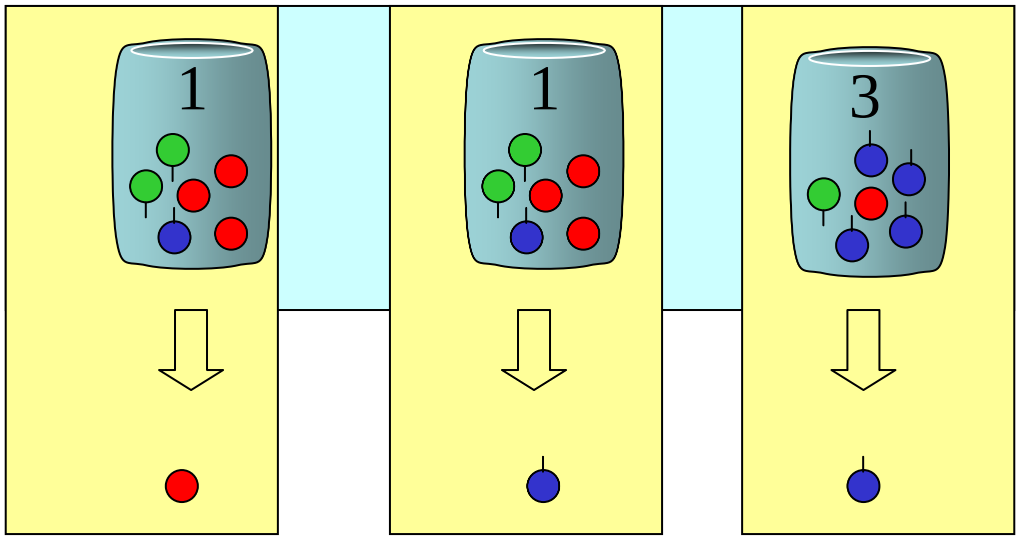


- *N* pots containing color balls
- *M* distinct colors
- Each pot contains different number of color balls

HMM: Generation Process

- Sequence generating algorithm
  - Step 1: Pick initial pot according to some random process
  - Step 2: Randomly pick a ball from the pot and then replace it
  - Step 3: Select another pot according to a random selection process
  - Step 4: Repeat steps 2 and 3



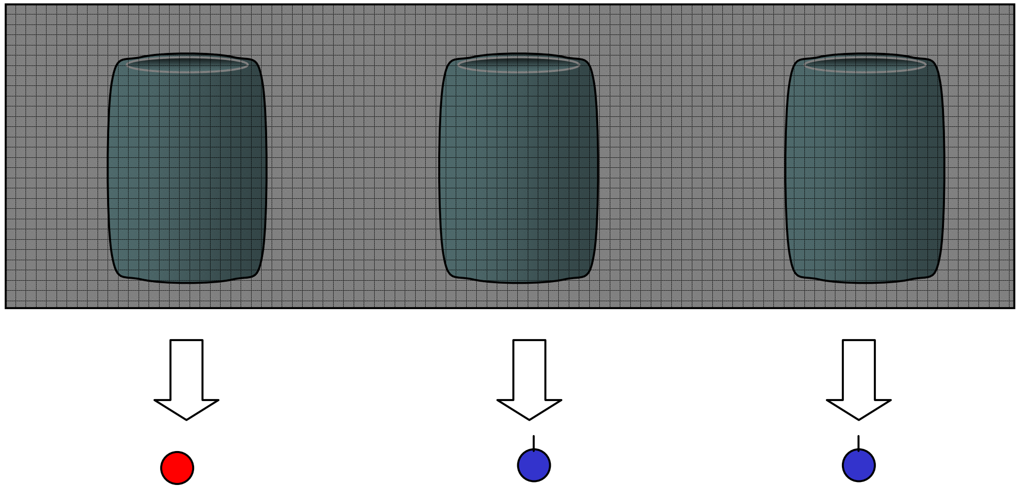


Markov process: $\{q(t)\}$

Output process: $\{f(x|q)\}$

HMM: Hidden Information

- Now, what is hidden?



  - We can just see the chosen balls
  - We can't see which pot is selected at a time
  - So, pot selection (state transition) information is hidden

HMM: Formal Definition

- Notation: $\lambda = (\mathrm{A}, \mathrm{B}, \pi)$
  (1) *N* : Number of states
  (2) *M* : Number of symbols observable in states
  $$V = \{v_1, \cdots, v_M\}$$
  (3) *A* : State transition probability distribution
  $$A = \{a_{ij}\}, \qquad 1 \le i, j \le N$$
  (4) *B* : Observation symbol probability distribution
  $$B = \{b_i(v_k)\}, \qquad 1 \le i \le N, 1 \le j \le M$$
  (5) π : Initial state distribution
  $$\pi_i = P(q_1 = i), \qquad 1 \le i \le N$$

Three Problems

1. Model evaluation problem
   – What is the probability of the observation?
   – Forward algorithm

2. Path decoding problem
   – What is the best state sequence for the observation?
   – Viterbi algorithm

3. Model training problem
   – How to estimate the model parameters?
   – Baum-Welch reestimation algorithm

Solution to
Model Evaluation Problem

Forward algorithm

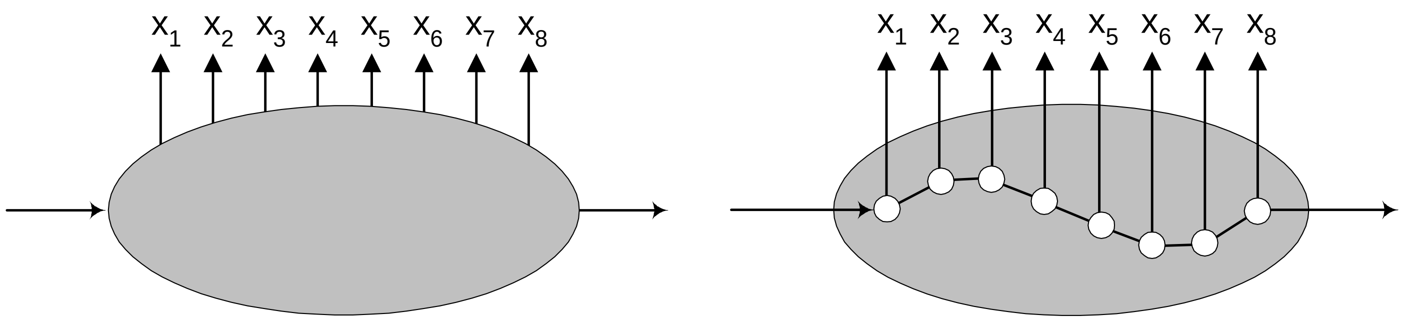
Backward algorithm

Definition

- Given a model λ
- Observation sequence: $X = x_1, x_2, \cdots, x_T$
- $P(X|\lambda) = ?$

- $P(X \mid \lambda) = \sum_{Q} P(X, Q \mid \lambda) = \sum_{Q} P(X \mid Q, \lambda) P(Q \mid \lambda)$

  (A path or state sequence: $Q = q_1, \cdots, q_T$)



Solution

- Easy but slow solution: exhaustive enumeration

$$P(X \mid \lambda) = \sum_Q P(X, Q \mid \lambda) = \sum_Q P(X \mid Q, \lambda) P(Q \mid \lambda)$$

$$= \sum_Q b_{q_1}(x_1) b_{q_2}(x_2) \cdots b_{q_T}(x_T) \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T}$$

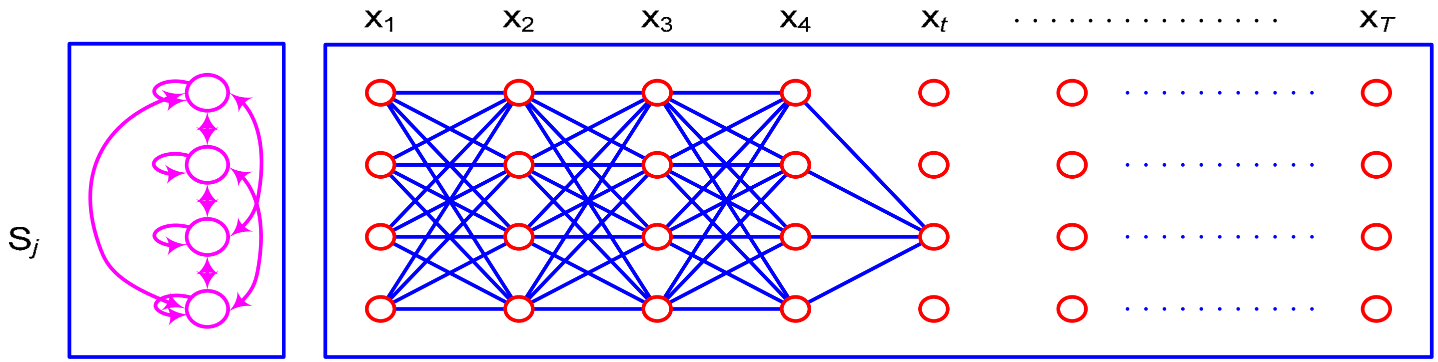
  - Exhaustive enumeration = combinational explosion!

$$O(N^T)$$

- Smart solution exists?
  - Yes!
  - Dynamic Programming technique
  - Lattice structure based computation
  - Highly efficient -- linear in frame length

Forward Algorithm

- Key idea
  – Span a lattice of *N* states and *T* times
  – Keep the sum of probabilities of all the paths coming to each state *i* at time *t*



- Forward probability

$$\alpha_t(j) = P(x_1 x_2 \ldots x_t, q_t = S_j \mid \lambda)$$

$$= \sum_{Q_t} P(x_1 x_2 \ldots x_t, Q_t = q_1 \ldots q_t \mid \lambda)$$

$$= \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(x_t)$$

Forward Algorithm

- Initialization

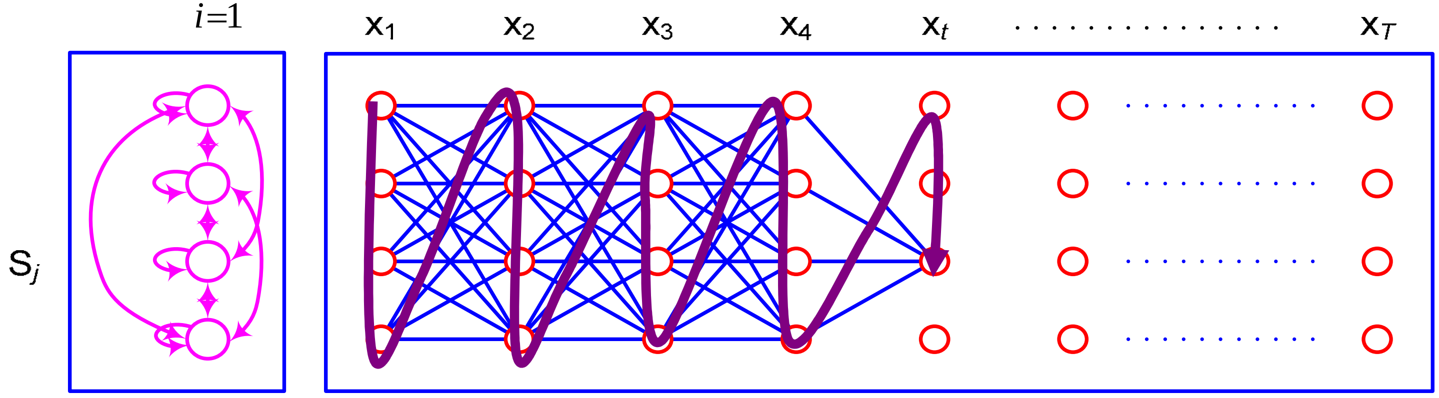
$$\alpha_1(i) = \pi_i b_i(\mathbf{x}_1) \qquad\qquad 1 \le i \le N$$

- Induction

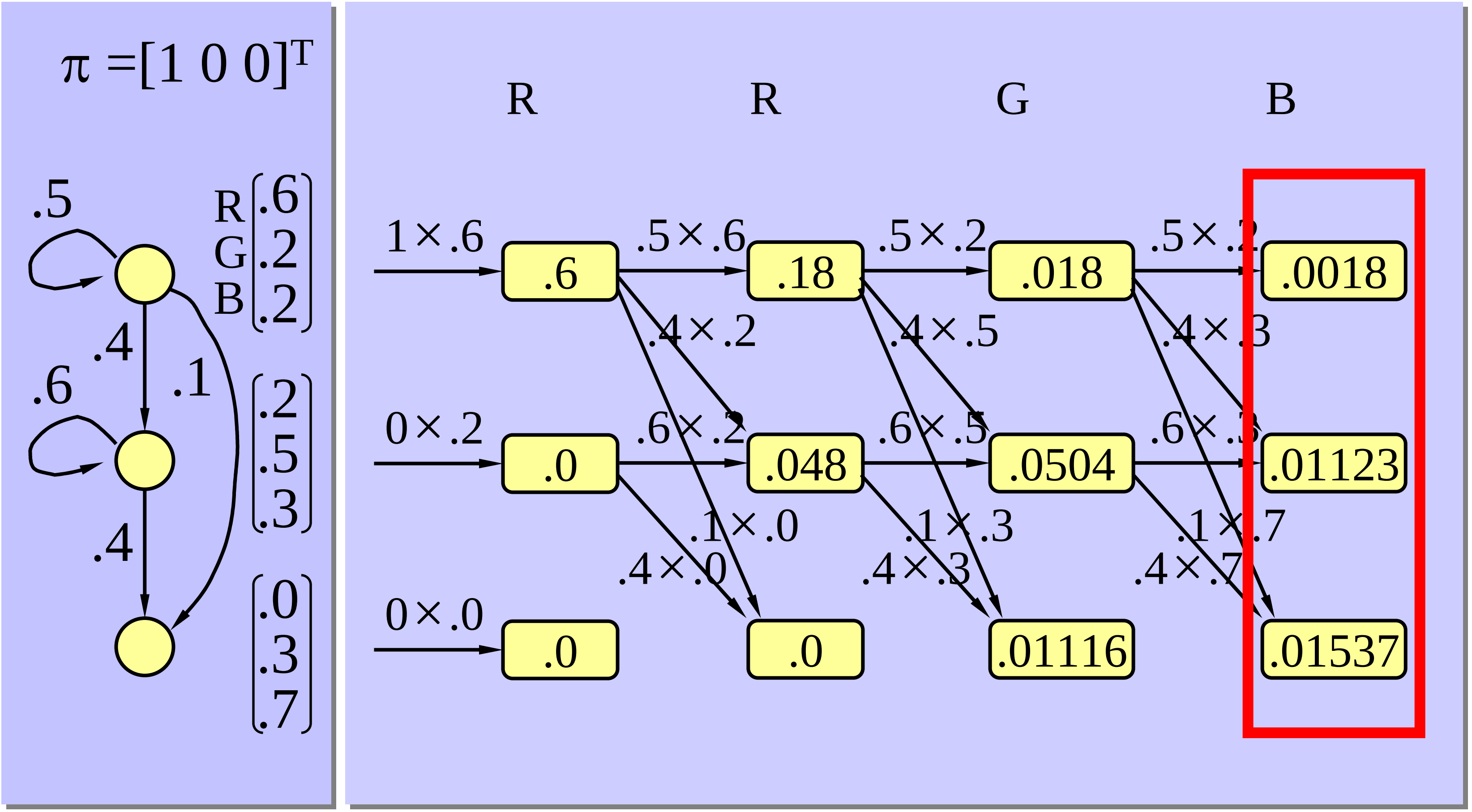
$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(\mathbf{x}_t) \qquad 1 \le j \le N,\ t = 2, 3, \cdots, T$$

- Termination

$$P(\mathbf{X} \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

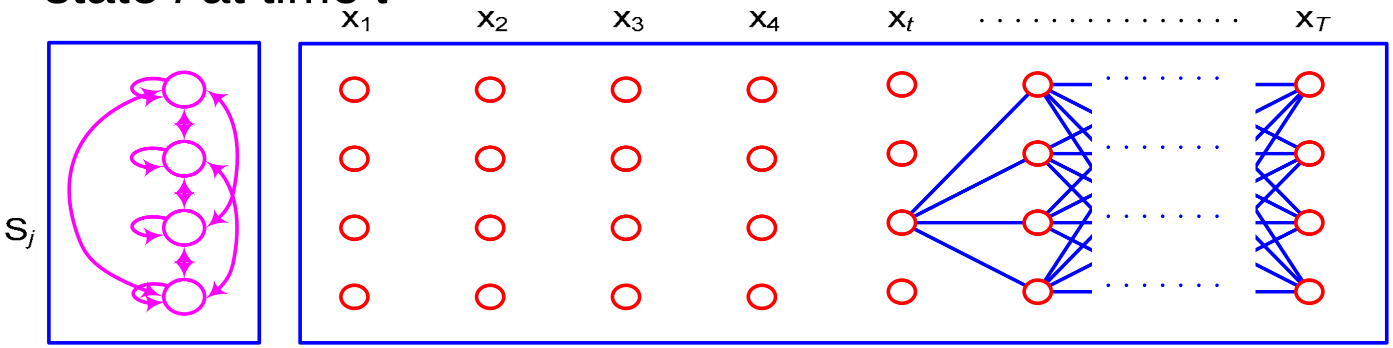


Numerical Example: $P(\mathrm{RRGB}|\lambda)$



Backward Algorithm (1)

- Key Idea
  – Span a lattice of *N* states and *T* times
  – Keep the sum of probabilities of all the outgoing paths at each state *i* at time *t*



- Backward probability

$$\beta_t(i) = P(x_{t+1}x_{t+2}...x_T \mid q_t = S_i, \lambda)$$

$$= \sum_{Q_{t+1}} P(x_{t+1}x_{t+2}...x_T, Q_{t+1} = q_{t+1}...q_T \mid q_t = S_i, \lambda)$$

$$= \sum_{j=1}^{N} a_{ij} b_j(x_{t+1}) \beta_{t+1}(j)$$

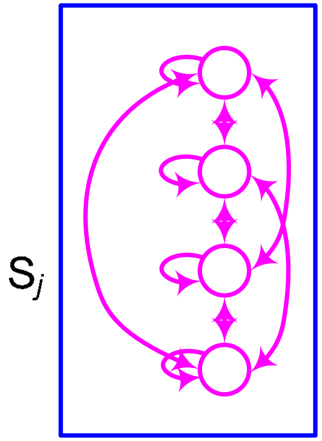
Backward Algorithm (2)

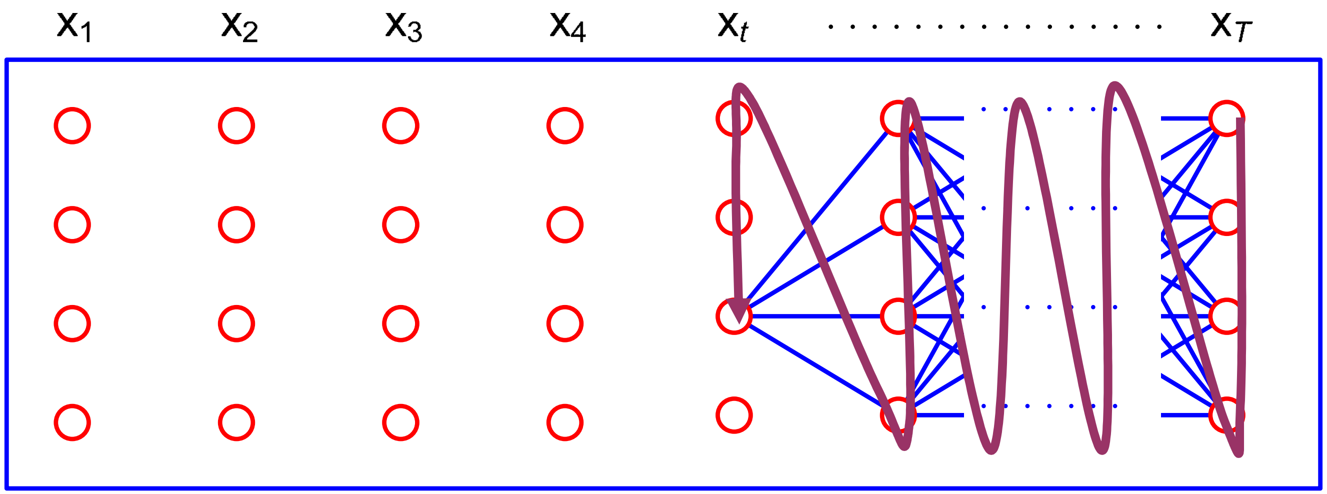
- Initialization

$$\beta_T(i) = 1 \qquad\qquad 1 \le i \le N$$

- Induction

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{x}_{t+1}) \beta_{t+1}(j) \qquad 1 \le i \le N, \quad t = T-1, T-2, \cdots, 1$$





Solution to
Path Decoding Problem

State sequence

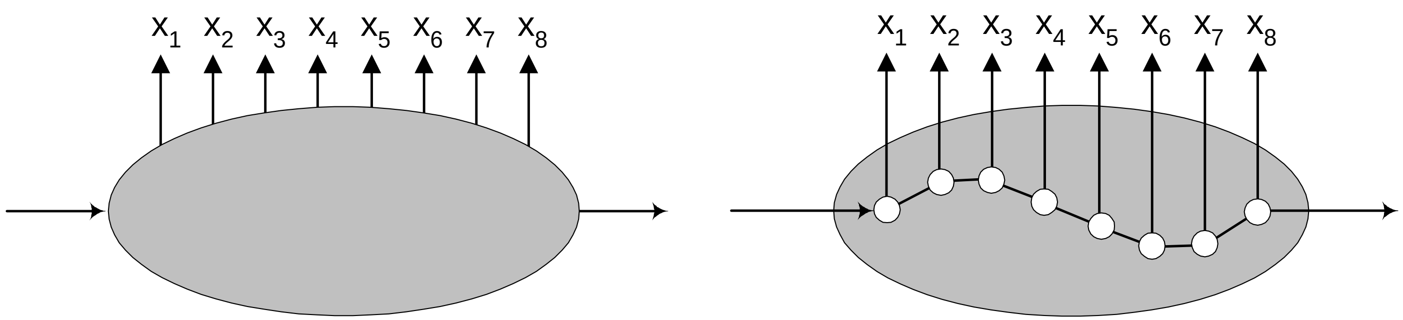
Optimal path

Viterbi algorithm

Sequence segmentation

The Most Probable Path

- Given a model λ
- Observation sequence: $X = \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T$
- $P(X, Q \mid \lambda)$ = ?
- $Q^* = \arg\max_Q P(X, Q \mid \lambda) = \arg\max_Q P(X \mid Q, \lambda) P(Q \mid \lambda)$
  - (A path or state sequence: $Q = q_1, \cdots, q_T$)



Viterbi Algorithm

- Purpose
  - An analysis for internal processing result
  - The best, the most likely state sequence
  - Internal segmentation

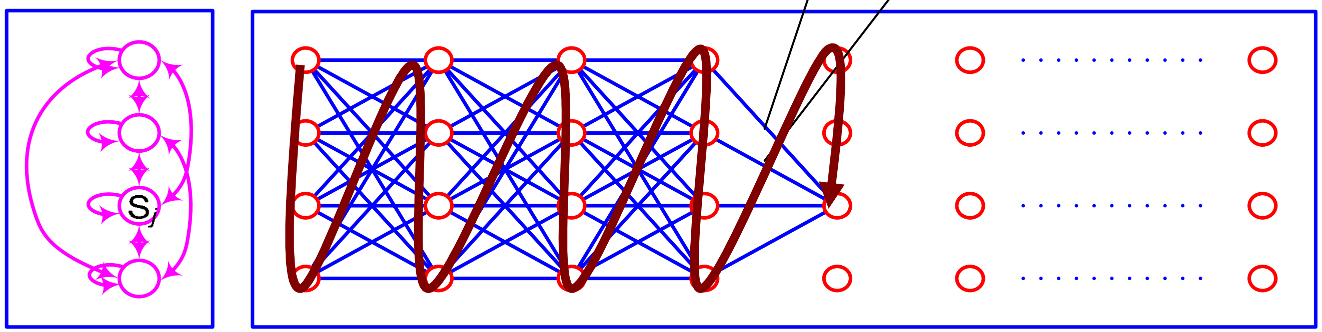
- Viterbi Algorithm
  - Alignment of observation and state transition
  - Dynamic programming technique

Viterbi Path Idea

- Key idea
  - Span a lattice of *N* states and *T* times
  - Keep the probability and the previous node of the most probable path coming to each state *i* at time *t*
- Recursive path selection
  - Path probability: $\delta_{t+1}(j) = \max_{1 \le i \le N} \delta_t(i) a_{ij} b_j(\mathbf{x}_{t+1})$
  - Path node: $\psi_{t+1}(j) = \arg\max_{1 \le i \le N} \delta_t(i) a_{ij}$

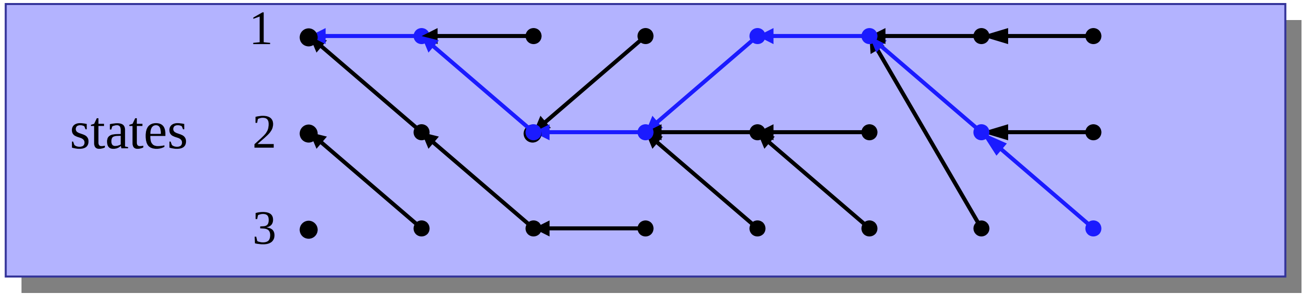
$\delta_t(1) a_{1j} b_j(\mathbf{x}_{t+1})$

$\delta_t(i) a_{ij} b_j(\mathbf{x}_{t+1})$

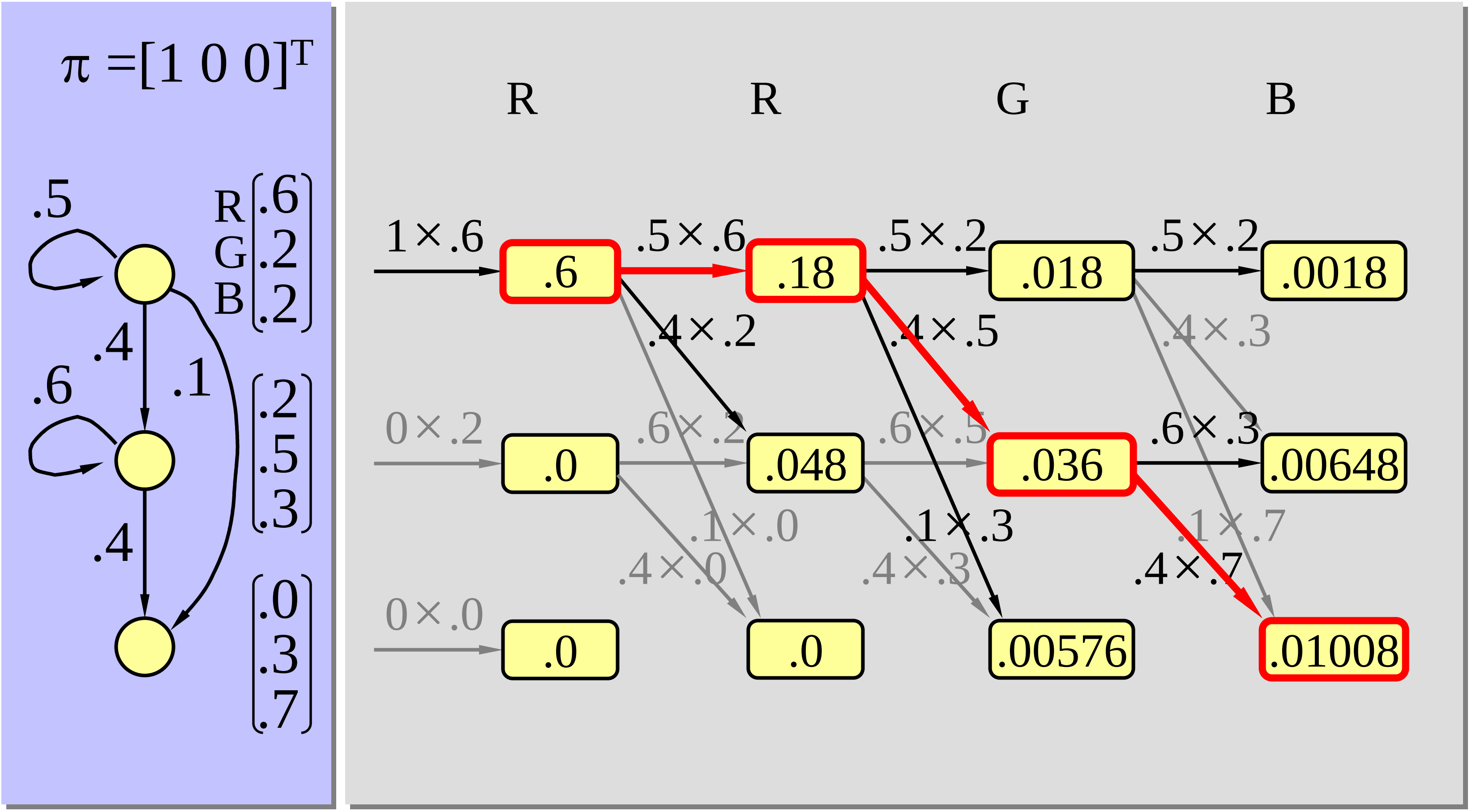


Viterbi Algorithm

- Introduction: $$\delta_1(i) = \pi_i b_i(\mathbf{x}_1), \quad 1 \le i \le N$$ $$\psi_1(i) = 0$$
- Recursion: $$\delta_{t+1}(j) = \max_{1 \le i \le N} \delta_t(i) a_{ij} b_j(\mathbf{x}_{t+1}), \quad 1 \le t \le T-1$$ $$\psi_{t+1}(j) = \arg\max_{1 \le i \le N} \delta_t(i) a_{ij} \qquad 1 \le j \le N$$
- Termination: $$P^* = \max_{1 \le i \le N} \delta_T(i)$$ $$q_T^* = \arg\max_{1 \le i \le N} \delta_T(i)$$
- Path backtracking: $$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, \ldots, 1$$



Numerical Example: P(RRGB,Q*|λ)



Solution to
Model training Problem

HMM training algorithm

Maximum likelihood estimation

Baum-Welch reestimation

HMM Training Algorithm

- Given an observation sequence $X = \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T$
- Find the model parameter $\lambda^* = (A, B, \pi)$

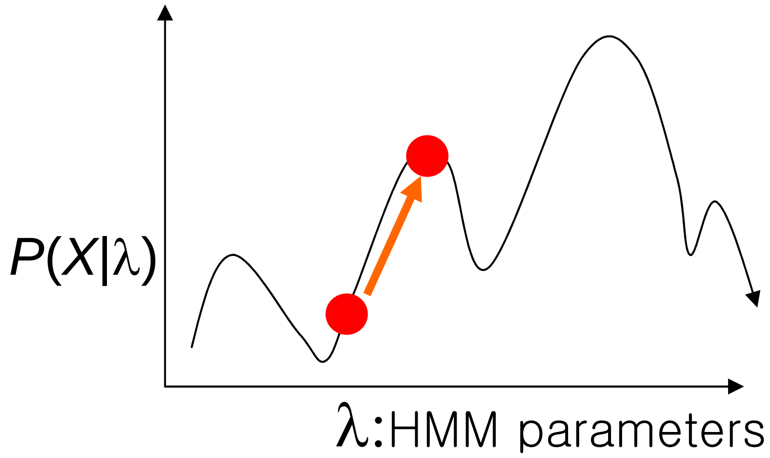
  s.t. $P(X \mid \lambda^*) \geq P(X \mid \lambda) \text{ for } \forall \lambda$

  - Adapt HMM parameters maximally to training samples
  - Likelihood of a sample

$$P(X \mid \lambda) = \sum_Q P(X \mid Q, \lambda) \boxed{P(Q \mid \lambda)}$$

State transition is hidden!

- NO analytical solution
- *Baum-Welch* reestimation (EM)
  - iterative procedures that locally maximizes $P(X|\lambda)$
  - convergence proven
  - MLE statistic estimation



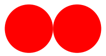
Maximum Likelihood Estimation

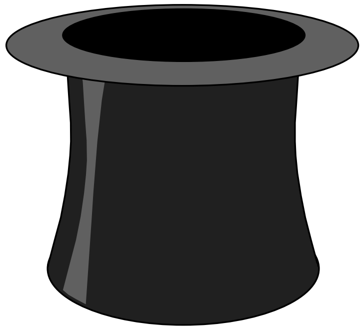
- MLE "selects those parameters that maximizes the probability function of the observed sample."

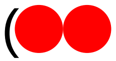
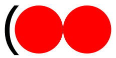
- [Definition] Maximum Likelihood Estimate
  - Θ: a set of distribution parameters
  - Given X, Θ^* is maximum likelihood estimate of Θ if

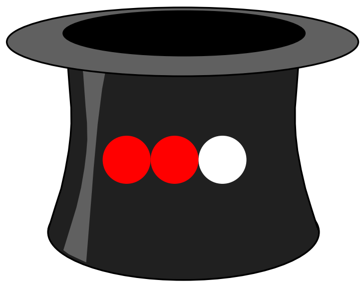
$$f(X|\Theta^*) = \max_{\Theta} f(X|\Theta)$$

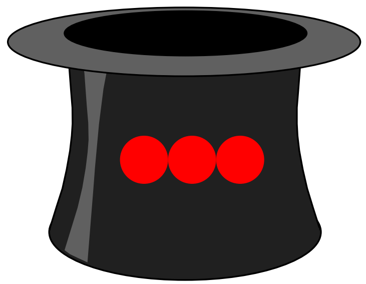
MLE Example

- Scenario
  - Known: 3 balls inside pot (some red; some white)
  - Unknown: R = # red balls
  - Observation:  (two reds)



- Two models
  - P ( |R=2) = $\binom{2}{2}\binom{1}{0}\Big/\binom{3}{2} = \frac{1}{3}$
  - P ( |R=3) = $\binom{3}{2}\Big/\binom{3}{2} = 1$





- Which model?
  - $L(\lambda_{R=3}) > L(\lambda_{R=2})$
  - Model(R=3) is our choice

MLE Example (Cont.)

- Model(R=3) is a more likely strategy,
  unless we have *a priori* knowledge of the system.
- However, without an observation of two red balls
  - No reason to prefer $P(\lambda_{R=3})$ to $P(\lambda_{R=2})$

- ML method chooses the set of parameters that maximizes the likelihood of the given observation.
- It makes parameters maximally adapted to training data.

EM Algorithm for Training

- With $\lambda^{(t)} =< \{a_{ij}\}, \{b_{ik}\}, \pi_i >$, estimate EXPECTATION of following quantities:
  - –Expected number of state *i* visiting
  - –Expected number of transitions from *i* to *j*

- With following quantities:
  - –Expected number of state *i* visiting
  - –Expected number of transitions from *i* to *j*
- Obtain the MAXIMUM LIKELIHOOD of

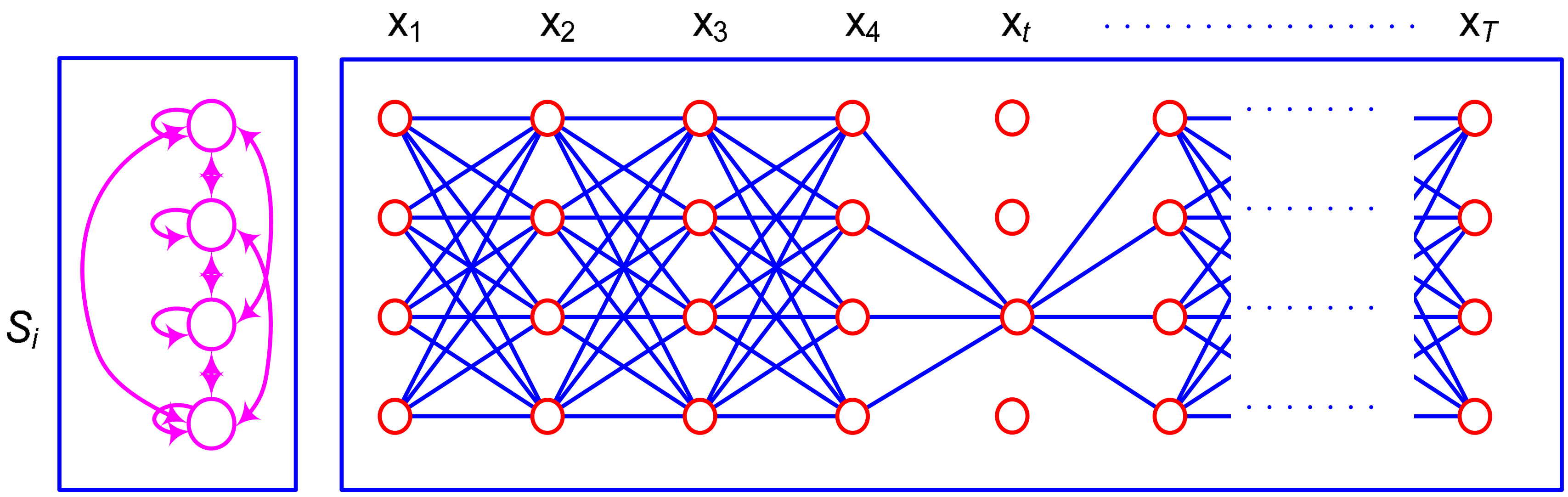
$$\lambda^{(t+1)} =< \{a'_{ij}\}, \{b'_{ik}\}, \pi'_i >$$

Expected Number of S_i Visiting

$$\gamma_t(i) = P(q_t = S_i \mid X, \lambda)$$

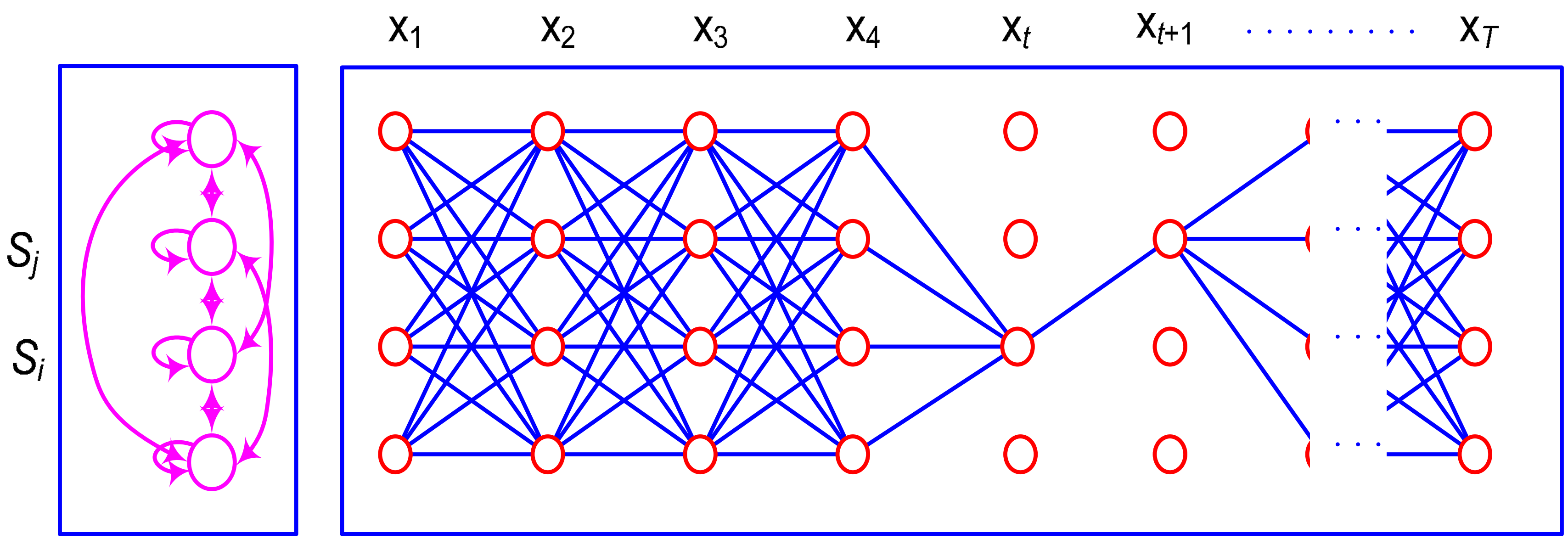
$$= \frac{P(q_t = S_i, X \mid \lambda)}{P(X \mid \lambda)}$$

$$= \frac{\alpha_t(i)\beta_t(i)}{\sum_j \alpha_t(j)\beta_t(j)}$$



Expected Number of Transition

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j \mid X, \lambda) = \frac{\alpha_t(i) a_{ij} b_j(x_{t+1}) \beta_{t+1}(j)}{\sum_i \sum_j \alpha_i(i) a_{ij} b_j(x_{t+1}) \beta_{t+1}(j)}$$



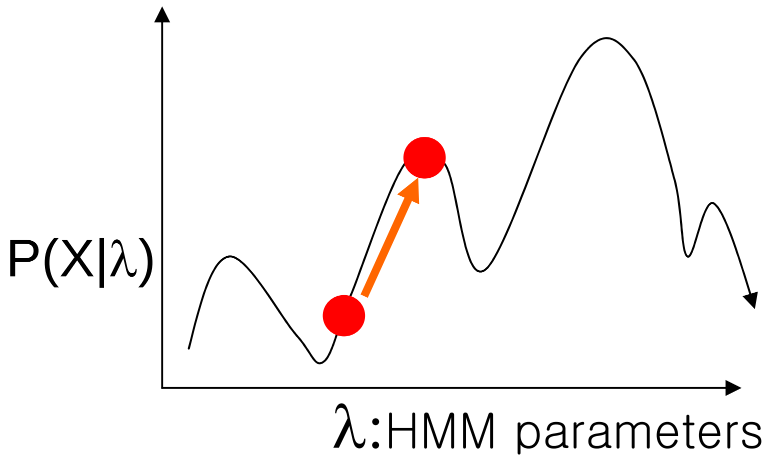
Parameter Reestimation

- MLE parameter estimation

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\bar{b}_j(v_k) = \frac{\sum_{\substack{t=1 \\ s.t.\ x_t = v_k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

$$\bar{\pi}_i = \gamma_1(i)$$



  - Iterative: $P(X \mid \lambda^{(t+1)}) \geq P(X \mid \lambda^{(t)})$
  - convergence proven:
  - arriving local optima

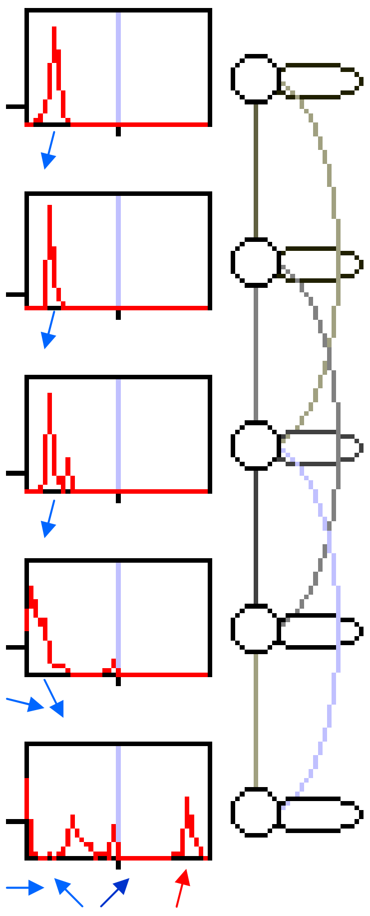
Other issues

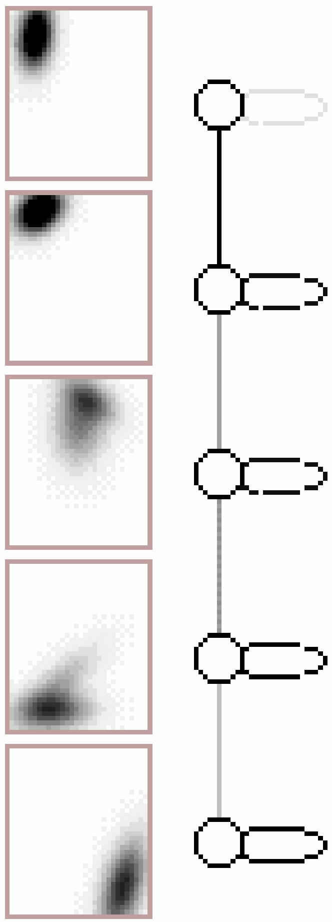
- Other method of training
  - MAP (Maximum A Posteriori) estimation – for adaptation
  - MMI (Maximum Mutual Information) estimation
  - MDI (Minimum Discrimination Information) estimation
  - Viterbi training
  - Discriminant/reinforcement training

- Other types of parametric structure
  - Continuous density HMM (CHMM)
    - More accurate, but much more parameters to train
  - Semi-continuous HMM
    - Mix of CHMM and DHMM, using parameter sharing
  - State-duration HMM
    - More accurate temporal behavior
- Other extensions
  - HMM+NN, Autoregressive HMM
  - 2D models: MRF, Hidden Mesh model, pseudo-2D HMM

Graphical DHMM and CHMM

- Models for ‘5’ and ‘2’





Pattern Classification using HMMs

- Pattern classification
- Extension of HMM structure
- Extension of HMM training method
- Practical issues of HMM
- HMM history

Pattern Classification

- Construct one HMM per each class *k*
  - $\lambda_1, \cdots, \lambda_N$
- Train each HMM λ_k with samples D_k
  - Baum-Welch reestimation algorithm

- Calculate model likelihood of $\lambda_1, \cdots, \lambda_N$ with observation X
  - Forward algorithm: $P(X \mid \lambda_k)$
- Find the model with maximum *a posteriori* probability

$$\begin{aligned}
\lambda^* &= \operatorname{argmax}_{\lambda_k} \; P(\lambda_k \mid X) \\
&= \operatorname{argmax}_{\lambda_k} \frac{P(\lambda_k) P(X \mid \lambda_k)}{P(X)} \\
&= \operatorname{argmax}_{\lambda_k} P(\lambda_k) P(X \mid \lambda_k)
\end{aligned}$$

Extension of HMM Structure

- Extension of state transition parameters
  - Duration modeling HMM
    - More accurate temporal behavior
  - Transition-output HMM
    - HMM output functions are attached to transitions rather than states
- Extension of observation parameter
  - Segmental HMM
    - More accurate modeling of trajectories at each state, but more computational cost
  - Continuous density HMM (CHMM)
    - Output distribution is modeled with mixture of Gaussian
  - Semi-continuous HMM (Tied mixture HMM)
    - Mix of continuous HMM and discrete HMM by sharing Gaussian components

Extension of HMM Training Method

- Maximum Likelihood Estimation (MLE)*
  - maximize the probability of the observed samples

- Maximum Mutual Information (MMI) Method
  - information-theoretic measure
  - maximize average mutual information:

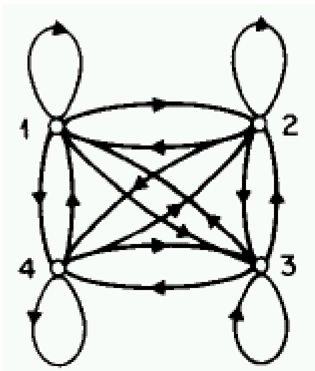
$$I^* = \max_{\lambda} \left\{ \sum_{v=1}^{V} \left[\log P(X^v \mid \lambda_v) - \log \sum_{w=1}^{V} P(X^w \mid \lambda_w) \right] \right\}$$

  - maximize discrimination power by training models together

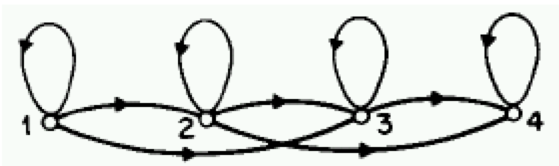
- Minimum Discrimination Information (MDI) Method
  - minimize the DI or the cross entropy between pd(signal) and pd(HMM)'s
  - use generalized Baum algorithm

Practical Issues of HMM

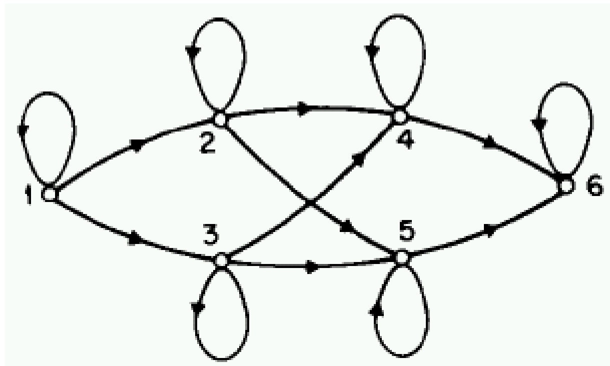
- Architectural and behavioral choices
  - the unit of modeling -- design choice
  - type of models: ergodic, left-right, parallel path.
  - number of states
  - observation symbols;discrete, continuous; mixture number
- Initial estimates
  - *A*, π : adequate with random or uniform initial values
  - *B* : good initial estimates are essential for CHMM



ergodic



left-right



parallel path

Practical Issues of HMM (Cont.)

- Scaling

$$\alpha_t(i) = \prod_{s=1}^{t-1} a_{s,s+1} \prod_{s=1}^{t} b_s(x_s)$$

  - heads exponentially to zero: → scaling (or using log likelihood)
- Multiple observation sequences
  - accumulate the expected freq. with weight $P(X(k)|l)$
- Insufficient training data
  - deleted interpolation with desired model & small model
  - output prob. smoothing (by local perturbation of symbols)
  - output probability tying between different states

Practical Issues of HMM (Cont.)

- HMM topology optimization
  - What to optimize
    - # of states
    - # of Gaussian mixtures per state
    - Transitions
  - Methods
    - Heuristic methods
      - # of states from average (or mod) length of input frames
    - Split / merge
      - # of states from iterative split / merge
    - Model selection criteria
      - # of states and mixtures at the same time
      - ML (maximum likelihood)
      - BIC (Bayesian information criteria)
      - HBIC (HMM-oriented BIC)
      - DIC (Discriminative information criteria)
      - ..

HMM applications and Software

- On-line handwriting recognition
- Speech applications
- HMM toolbox for Matlab
- HTK (hidden Markov model Toolkit)

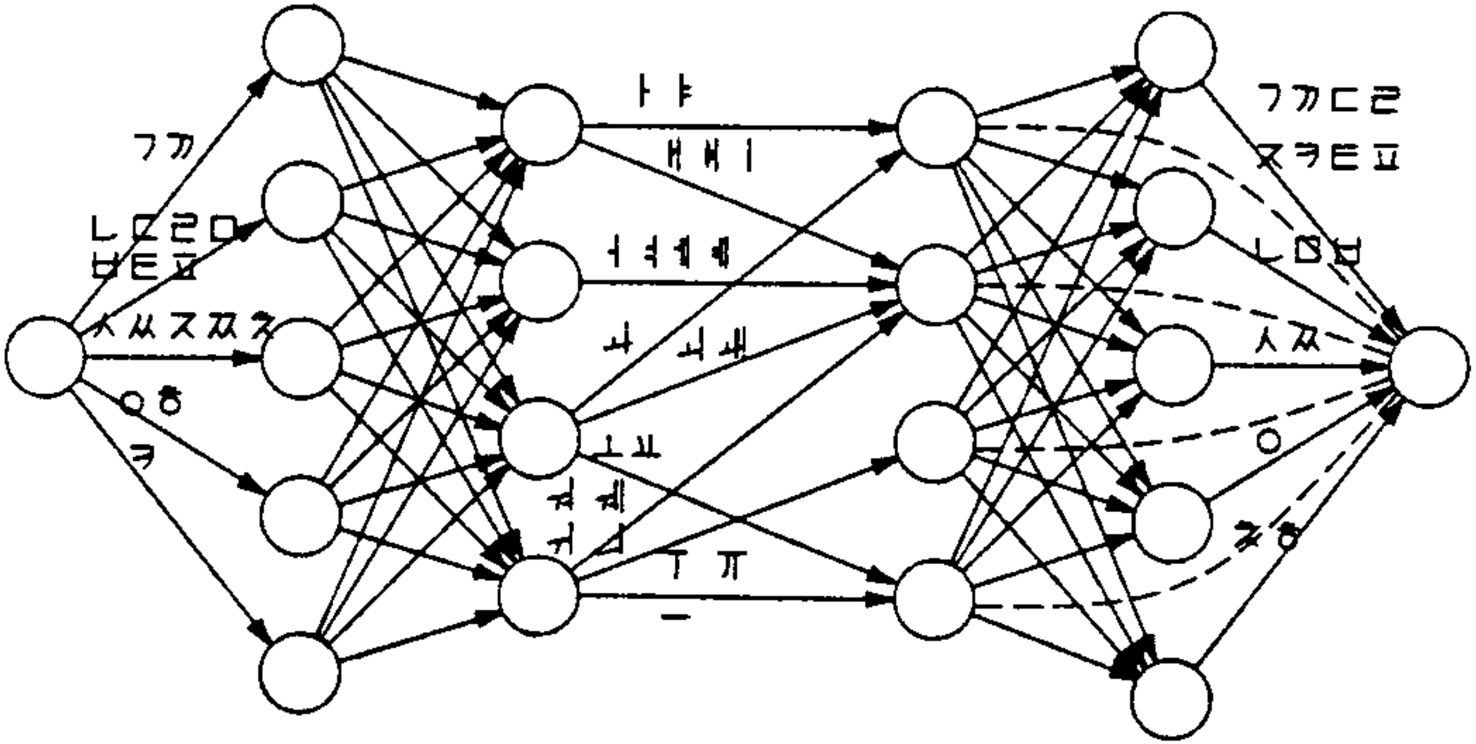
HMM Applications

- On-line handwriting recognition
  - BongNet: HMM network-based handwriting recognition system
- Speech applications
  - CMU Sphinx : Speech recognition toolkit
  - 언어과학 Dr.Speaking : English pronunciation correction system

BongNet

- Consortium of CAIR(Center for Artificial Intelligence Research) at KAIST
  - The name “BongNet” from its major inventor, BongKee Shin
- Prominent performance for unconstrained on-line Hangul recognition
- Modeling of Hangul handwriting
  - considers ligature between letters as well as consonants and vowels
    - (initial consonant)+(ligature)+(vowel)
    - (initial consonant)+(ligature)+(vowel)+(ligature)+(final consonant)
  - connects letter models and ligature models using Hangul composition principle
  - further improvements
    - BongNet+ : incorporating structural information explicitly
    - Circular BongNet : successive character recognition
    - Unified BongNet : Hangul and alphanumeric recognition
    - dictionary look-up

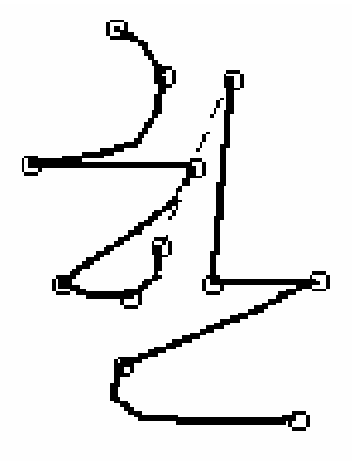
- Network structure



A Modification to BongNet

16-dir Chaincode ➔ Structure Code Generation

- Structure code sequence
  - carries structural information
    - not easily acquired using chain code sequence
    - including length, direction, and vending

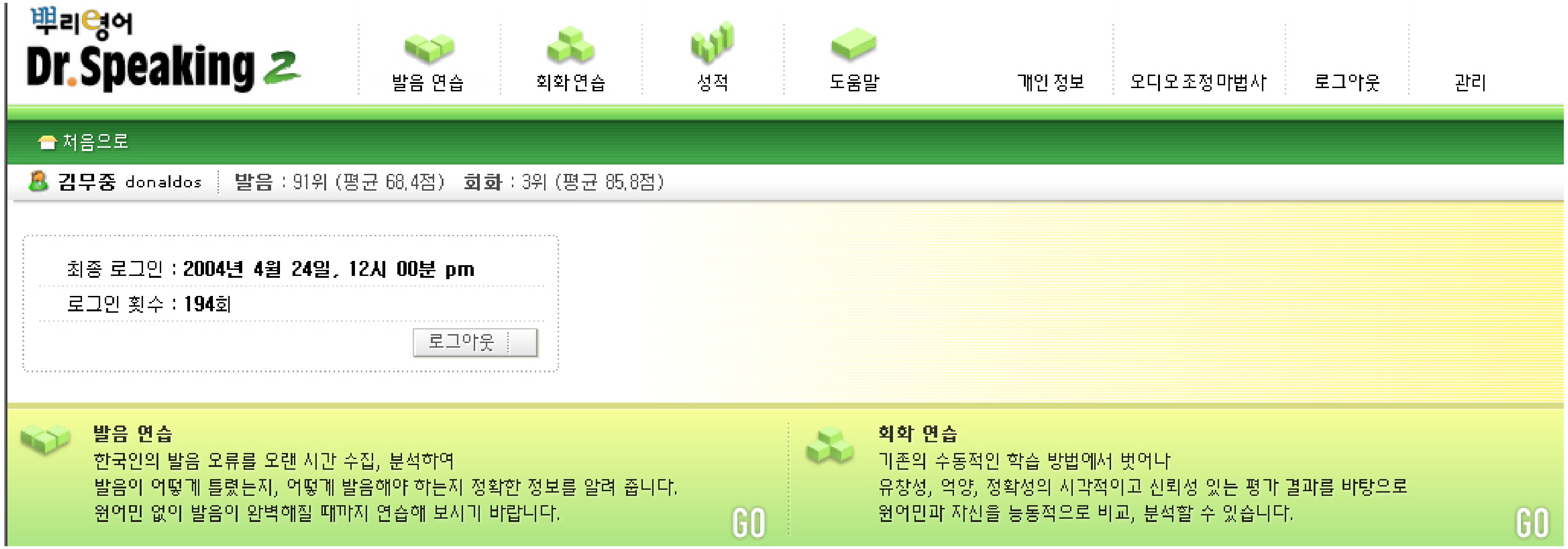


Distance Rotation	Straightness	Direction		Real
18.213	96.828	46.813	1	1
45.934	87.675	146.230	1	1
41.238	99.997	0.301	1	0
45.796	97.941	138.221	1	1
18.299	98.820	8.777	1	0
16.531	88.824	298.276	1	-1
45.957	100.000	293.199	0	0
52.815	99.999	95.421	1	0
26.917	99.961	356.488	1	0
53.588	99.881	156.188	1	0
56.840	80.187	17.449	1	-1

➡

3 37 0 37 0 31
54 11 28 15 5

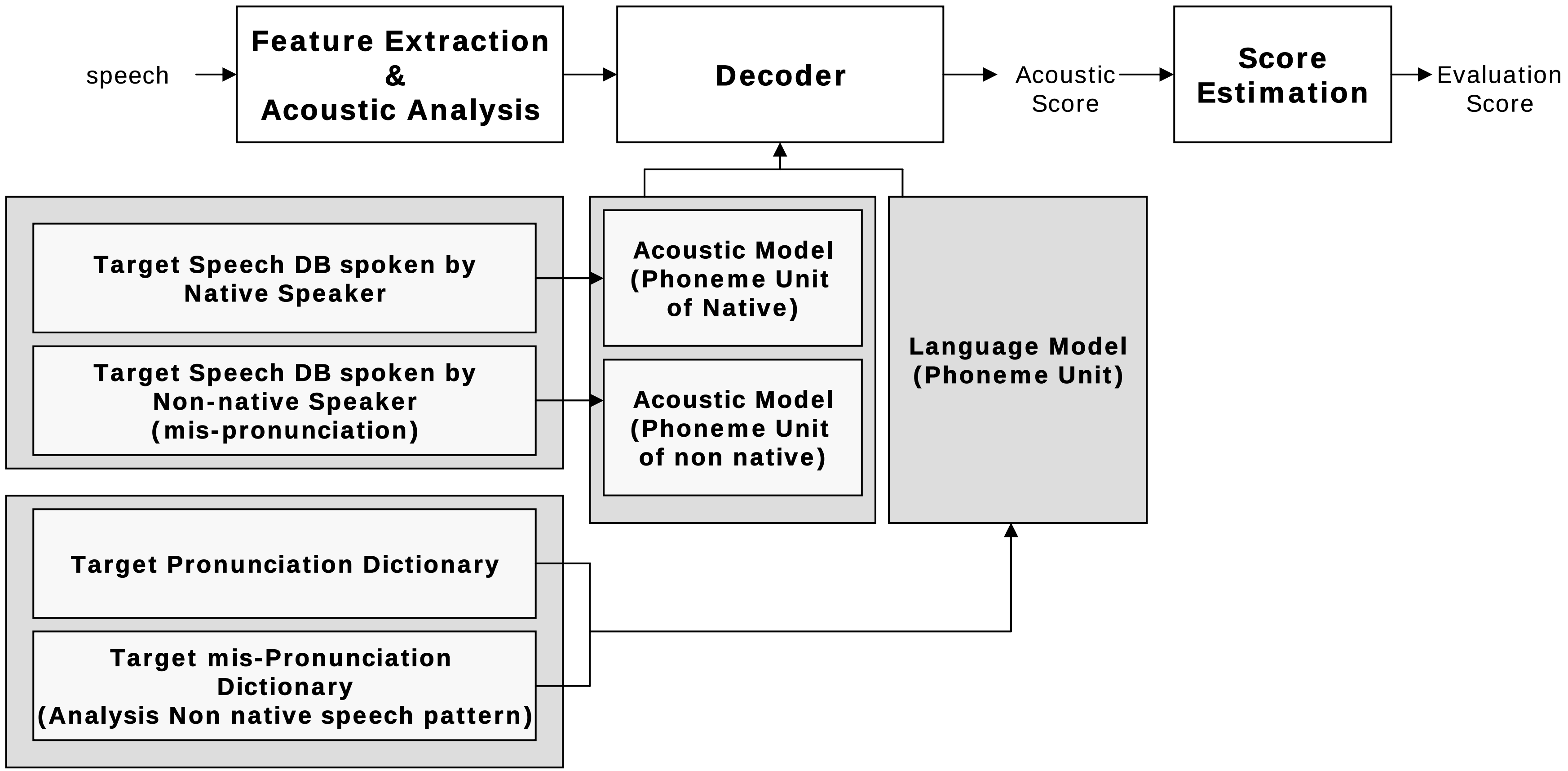
Dr. Speaking



1 단어수준에서 발음연습 – 음소단위 오류패턴 검출

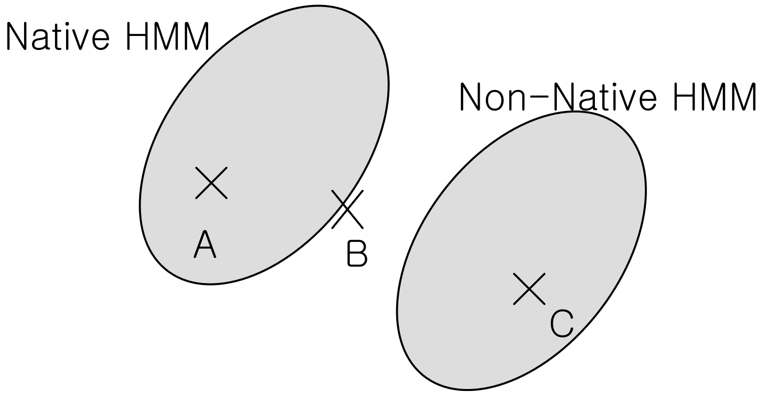
2 문장수준에서 발음연습 – 정확성, 유창성, 억양별 발음 평가

시스템 구조

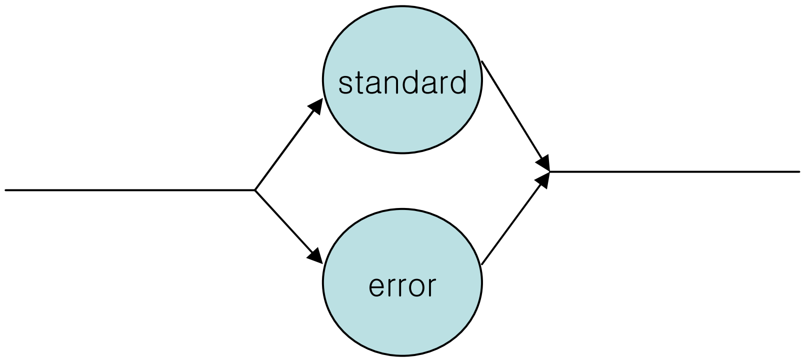


speech
Feature Extraction & Acoustic Analysis
Decoder
Acoustic Score
Score Estimation
Evaluation Score
Target Speech DB spoken by Native Speaker
Target Speech DB spoken by Non-native Speaker (mis-pronunciation)
Acoustic Model (Phoneme Unit of Native)
Acoustic Model (Phoneme Unit of non native)
Language Model (Phoneme Unit)
Target Pronunciation Dictionary
Target mis-Pronunciation Dictionary (Analysis Non native speech pattern)

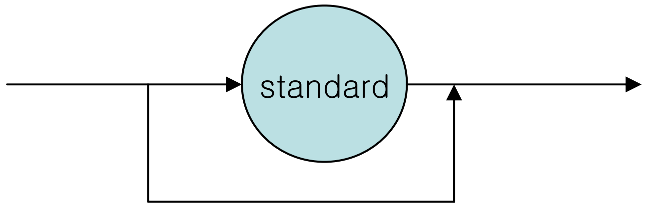
- Acoustic modeling



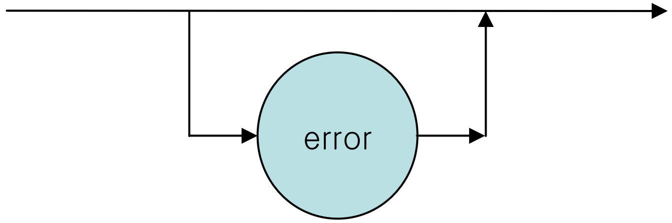
- Language modeling



replacement error modeling



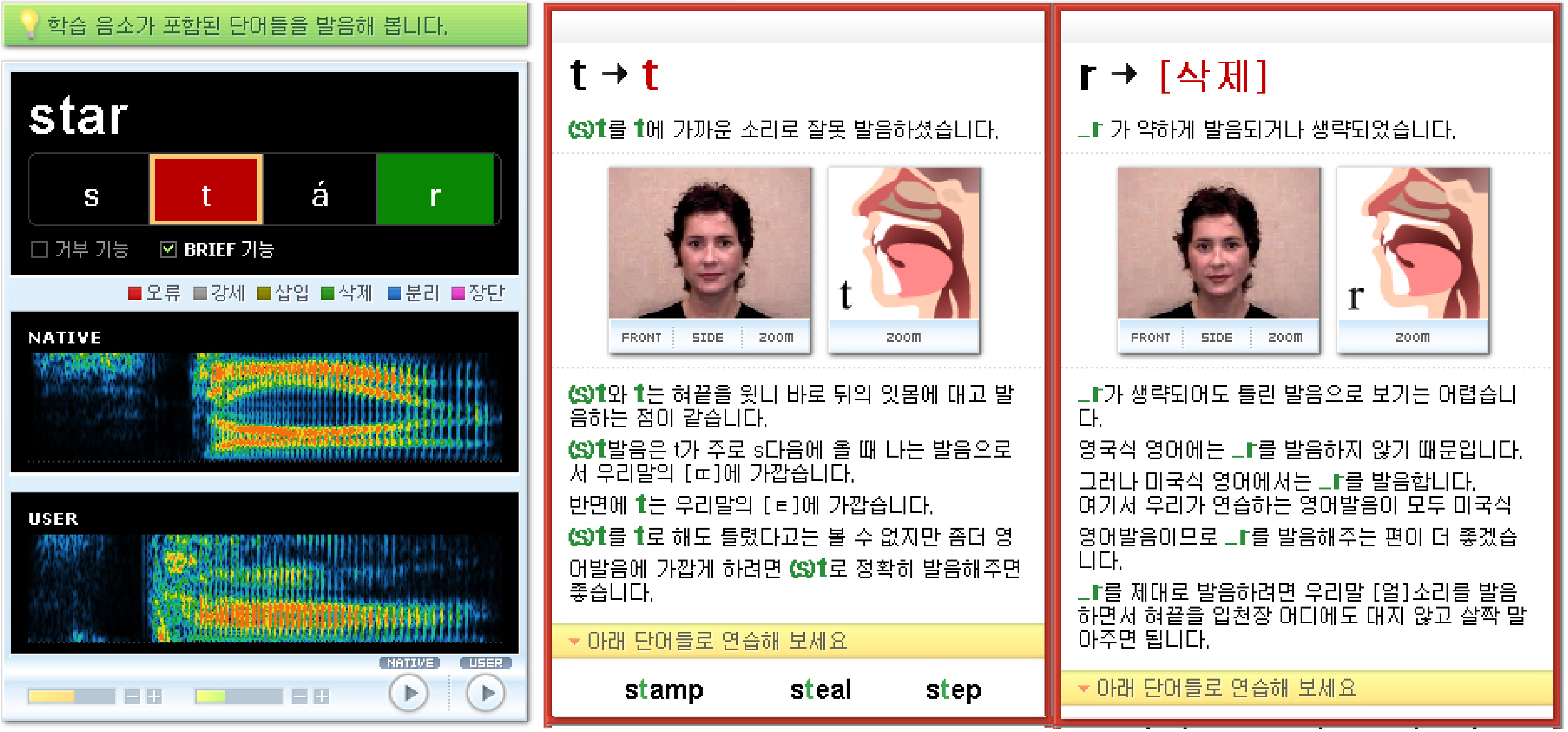
deletion error modeling



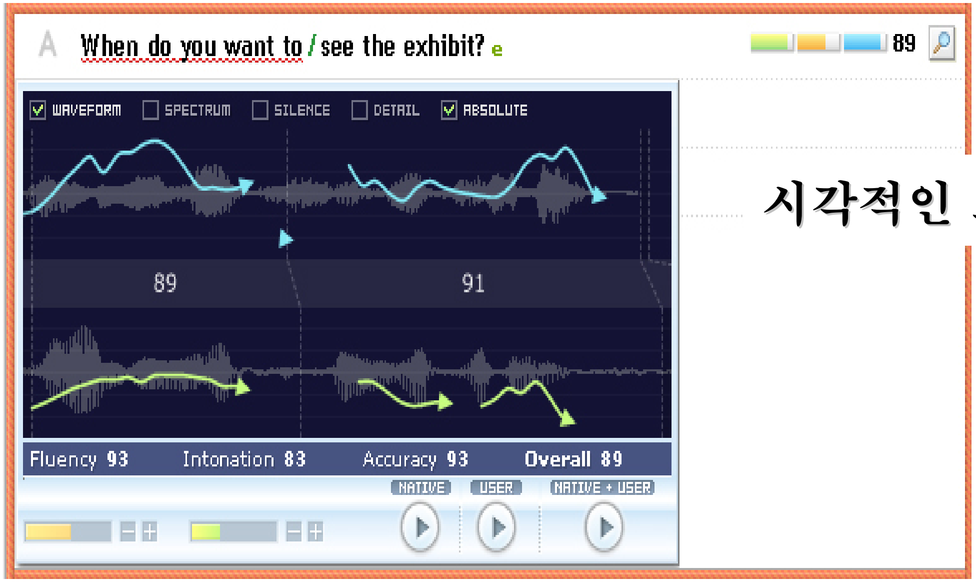
insertion error modeling

- **단어수준 발음 교정**

  : 음소단위 오류패턴 검출 – 오류발음대치, 삽입, 삭제, 이중모음 분리, 강세, 장단오류



- 문장 발음 연습



시각적인 교정정보 및 수준별 학습 평가

1 정확성평가 – 정발음패턴과 다양한 유형의 오류발음 패턴을 기반으로 평가

2 억양평가 – 억양관련 음성신호 추출 후 표준패턴과 오류패턴을 기반으로 평가

3 유창성평가 – 연음여부, 끊어 읽기, 발화구간 등 다양한 평가요소를 기반으로 평가

Software Tools for HMM

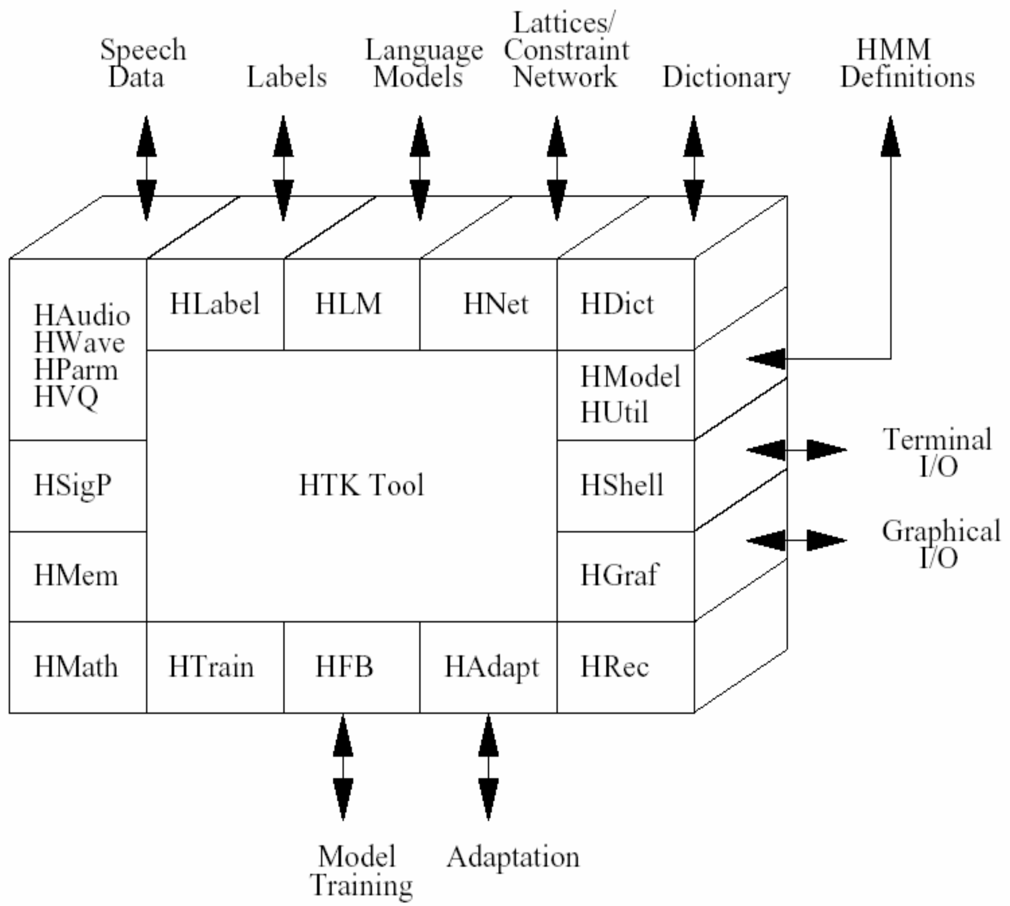
- **HMM toolbox for Matlab**
  - Developed by Kevin Murphy
  - Freely downloadable SW written in Matlab (Hmm… Matlab is not free!)
  - Easy-to-use: flexible data structure and fast prototyping by Matlab
  - Somewhat slow performance due to Matlab
  - Download: http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html

- HTK (Hidden Markov toolkit)
  - Developed by Speech Vision and Robotics Group of Cambridge University
  - Freely downloadable SW written in C
  - Useful for speech recognition research: comprehensive set of programs for training, recognizing and analyzing speech signals
  - Powerful and comprehensive, but somewhat complicate
  - Download: http://htk.eng.cam.ac.uk/

What is HTK ?

- Hidden Markov Model Toolkit
- Set of tools for training and evaluation HMMs
- Primarily used in automatic speech recognition and economic modeling
- Modular implementation, (relatively) easy to extend

HTK Software Architecture



- HShell : User input/output & interaction with the OS
- HLabel : Label files
- HLM : Language model
- HNet : Network and lattices
- HDic : Dictionaries
- HVQ : VQ codebooks
- HModel : HMM definitions
- HMem : Memory management
- HGraf : Graphics
- HAdapt : Adaptation
- HRec : main recognition processing functions

Generic Properties of a HTK Tool

- Designed to run with a traditional command-line style interface
- Each tool has a number of required argument plus optional arguments

```
HFoo  -T  1  -f  34.3  -a  -s  myfile  file1  file2
```

  - This tool has two main arguments called file1 and file2 plus four optional arguments
  - -f : real number, -T : integer, -s : string, -a : no following value

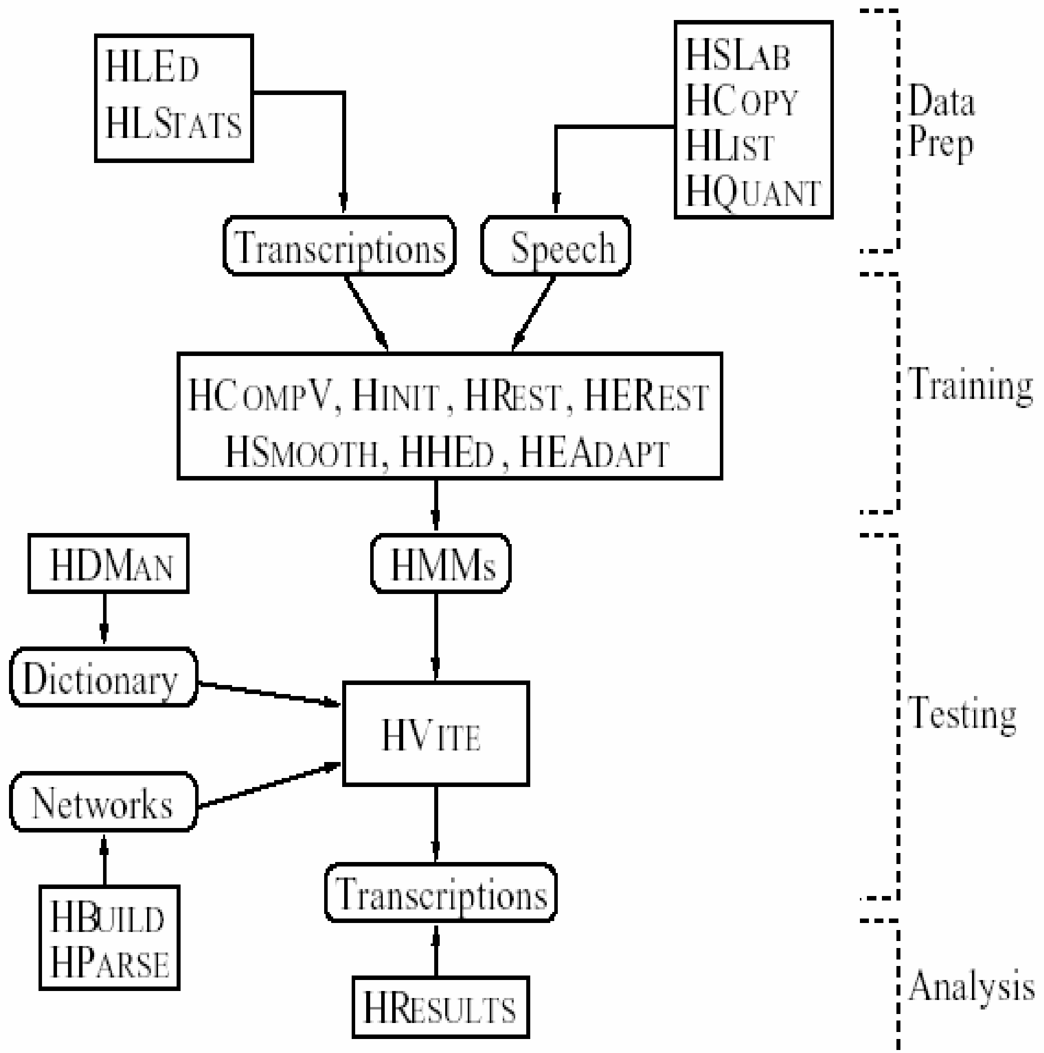
```
HFoo  -C  config  -f  34.3  -a  -s  myfile  file1  file2
```

  - HFoo will load the parameters stored in the configuration file config during its initialization procedures
  - Configuration parameters can sometimes by used as an alternative to using command line arguments

The Toolkit

- There are 4 main phases
  - data preparation, training, testing and analysis

- The Toolkit
  - Data Preparation Tools
  - Training Tools
  - Recognition Tools
  - Analysis Tools



< HTK Processing Stages >

Data Preparation Tools

- A set of speech data file and their associated transcriptions are required
- It must by converted into the appropriate parametric form

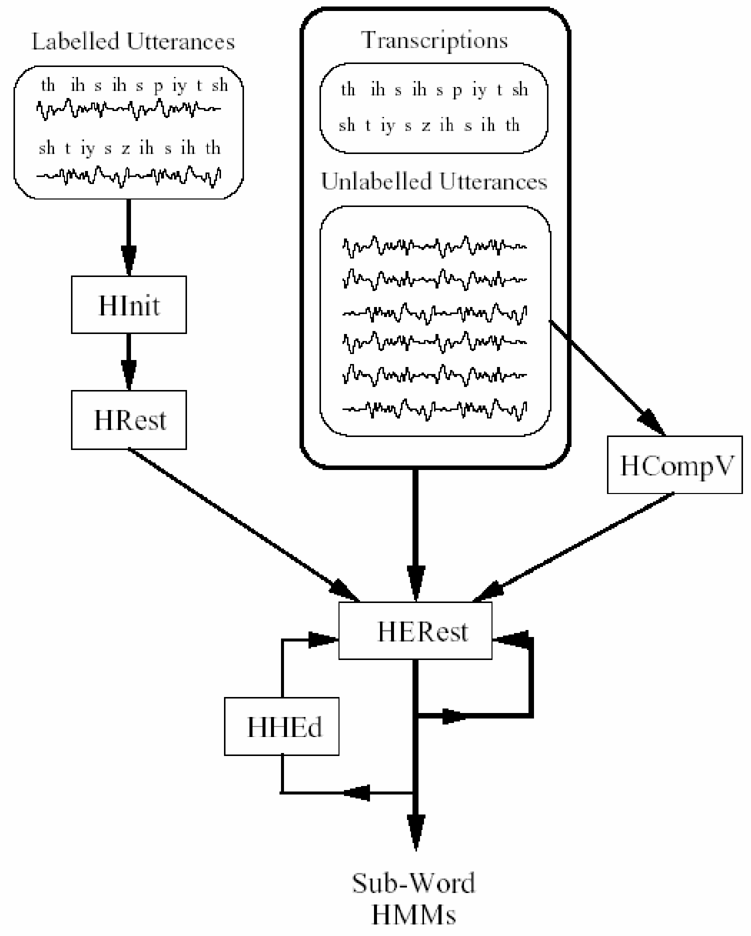
- **HSlab** : Used both to record the speech and to manually annotate it with and required transcriptions
- **HCopy** : simply copying each file performs the required encoding
- **HList** : used to check the contents of any speech file
- **HLed** : output file to a single *Master Label file* MLF which is usually more convenient for subsequent processing
- **HLstats** : gather and display statistics on label files and where required
- **HQuant** : used to build a VQ codebook in preparation for building discrete probability HMM system

Training Tools

- If there is some speech data available for which the location of the sub-word boundaries have been marked, this can be used as *bootstrap data*

- **HInit** and **HRest** provide isolated word style training using the fully labeled bootstrap data

- Each of the required HMMs is generated individually



Training Tools (cont'd)

- **HInit** : iteratively compute an initial set of parameter values using a *segmental k-means* procedure
- **HRest** : process fully labeled bootstrap data using a Baum-Welch re-estimation procedure
- **HCompV** : all of the phone models are initialized to by identical and have state means and variances equal to the global speech mean and variance
- **HERest** : perform a single Baum-Welch re-estimation of the whole set of HMM phone models simultaneously
- **HHed** : apply a variety of parameter tying and increment the number of mixture components in specified distributions
- **HEadapt** : adapt HMMs to better model the characteristics of particular speakers using a small amount of training or adaptation data

Recognition Tools

- **HVite** : use the token passing algorithm to perform Viterbi-based speech recognition
- **HBuild** : allow sub-networks to be created and used within higher level networks
- **HParse** : convert EBNF into the equivalent word network
- **HSgen** : compute the empirical perplexity of the task
- **HDman** : dictionary management tool

Analysis Tools

- **HResults**
  - Use dynamic programming to align the two transcriptions and count substitution, deletion and insertion errors
  - Provide speaker-by-speaker breakdowns, confusion matrices and time –aligned transcriptions
  - Compute *Figure of Merit* scores and *Receiver Operation Curve* information

HTK Example

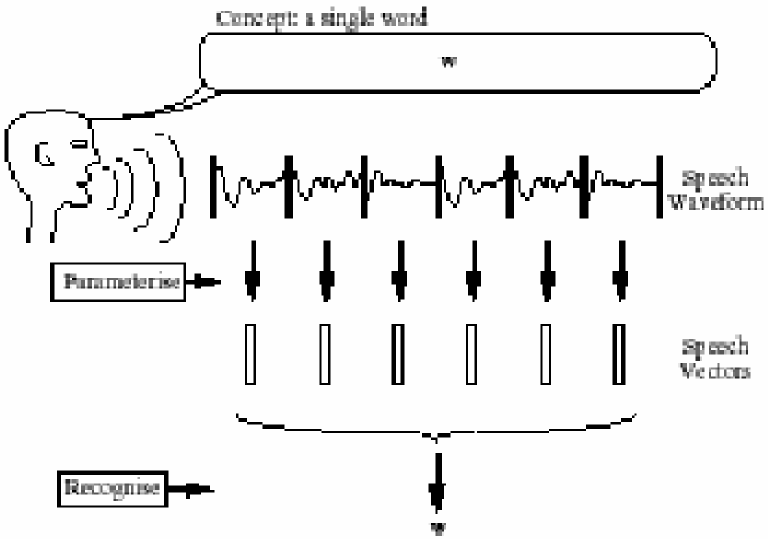
- Isolated word recognition

$$O = o_1, o_2, \cdots, o_T$$

spoken word be represented
by a sequence of vectors or *observations* O

$$\arg\max_i \{P(w_i \mid O)\}$$

Isolated word recognition
w_i : the i'th vocabulary word



$$\boxed{P(w_i \mid O)} = \frac{P(O \mid w_i) P(w_i)}{P(O)}$$

given set

depend on the likelihood $P(O|w_i)$

- Isolated word recognition (cont'd)

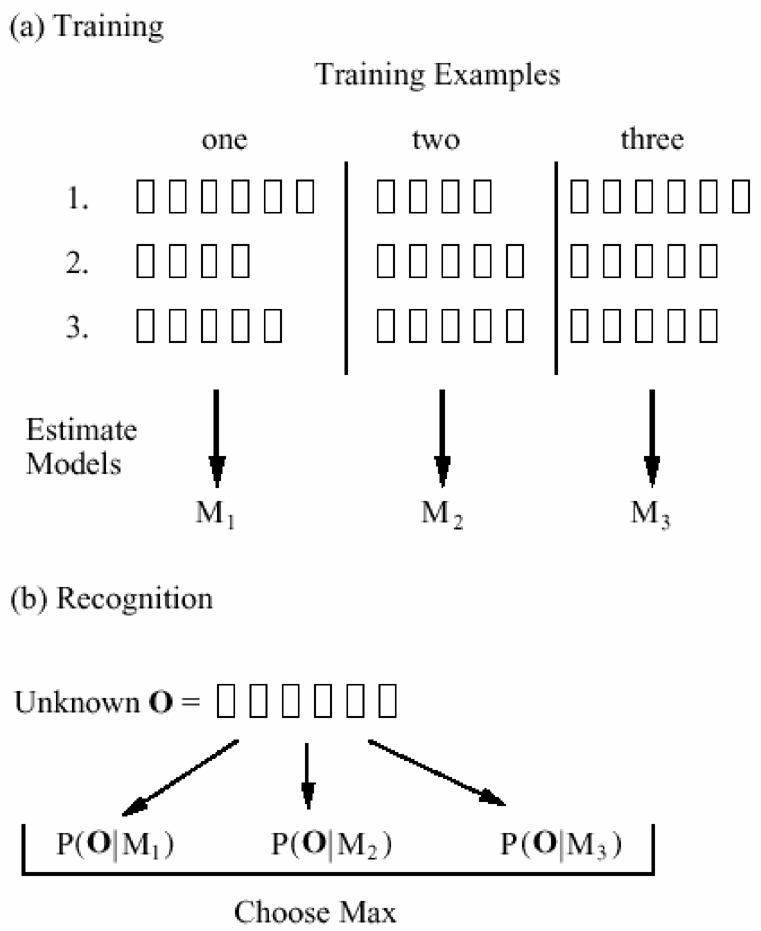
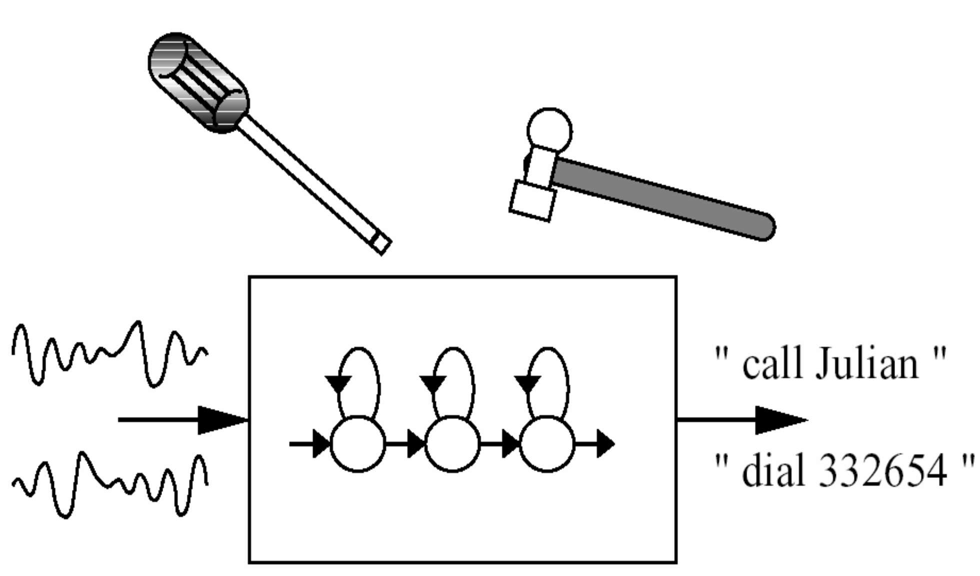


Fig. 1.4 Using HMMs for Isolated Word Recognition

Speech Recognition Example using HTK

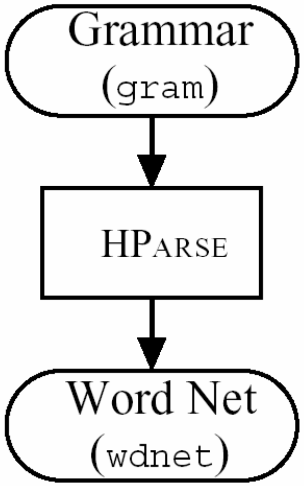
- Recognizer for voice dialing application
  - Goal of our system
    - Provide a voice-operated interface for phone dialing
  - Recognizer
    - digit strings, limited set of names
    - sub-word based



1> gram 파일을 생성한다.

- gram파일은 사용할 grammar를 정의한 파일로서
  전체적인 시나리오의 구성을 알려주는 파일이다.

```
----------------------- gram -------------------------
$digit = 일 | 이 | 삼 | 사 | 오 |..... | 구 | 공;
$name = 철수 | 만수 | ..... | 길동;
( SENT-START ( 누르기 <$digit> | 호출 $name) SENT-ENT )
-------------------------------------------------------
```



$표시 이후는 각 단어군의 정의이고 맨 아랫줄이 문법이다.

< >속의 내용은 반복되는 내용이며 |은 or(또는) 기호이다.

SENT-START 로 시작해서 SENT-END로 끝이 난다.

2> HParse gram wdnet 명령 실행.

- HParse.exe가 실행되어 gram파일로부터 wdnet을 생성시킨다.

3> dict 생성

- 단어 수준에서 각 단어의 음소를 정의 한다.

-------------------- dict --------------------------

SENT-END[] sil

SENT-START[] sil

공 kc oxc ngc sp

구 kc uxc sp

....

영희 jeoc ngc hc euic sp

....

팔 phc axc lc sp

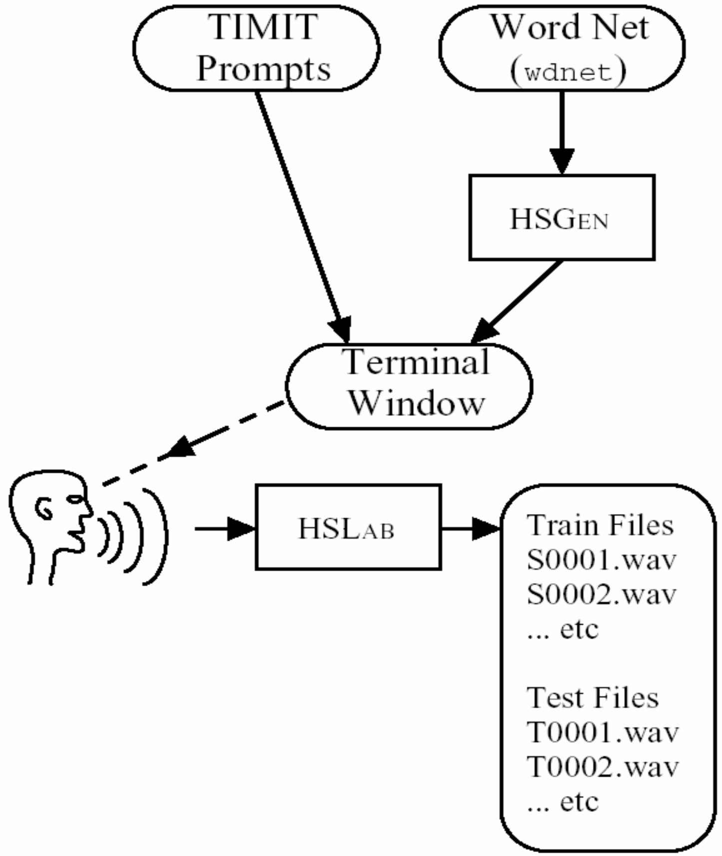
호출 hc oxc chc uxc lc sp

4> HSGen -l -n 200 wdnet dict
명령실행

- wdnet 과 dict를 이용하여 HSGen.exe가 실행되어 입력 가능한 문장 200개를 생성해 준다.

5> HSGen이 만들어준 훈련용 문장을 녹음한다.

- HSLab 또는 일반 녹음 툴 사용



6> words.mlf 파일을 작성한다.

- words.mlf 파일은 녹음한 음성 파일들의 전사파일의 모음이다.

```
--------------------- words.mlf ----------------------
#!MLF!#
"*/s0001.lab"
누르기
공
이
칠
공
구
일
.
#*/s0002.lab"
호출
영희
.
.....
------------------------------------------------------
```

7> mkphones0.led 파일의 작성

-mkphones0.led 은 words.mlf 파일의 각 단어를 음소로 치환시킬 때의 옵션들을 저장하는 파일이다.

------------- mkphones0.led ----------------
EX
IS sil sil
DE sp

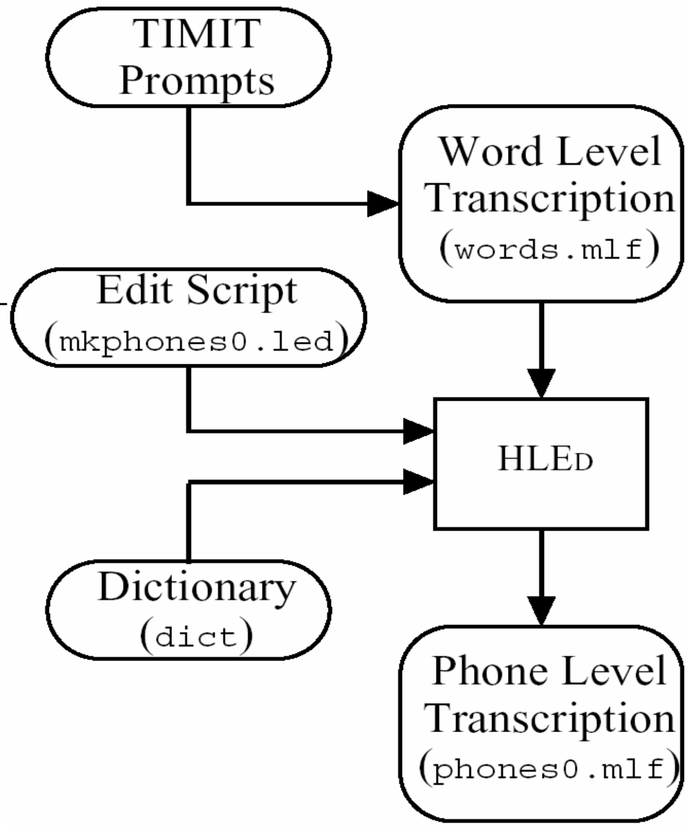
위의 옵션의 뜻은 문장의 양끝에 sil을 삽입하고 sp는 삭제한다는 의미.

8>HLEd -d dict -i phones0.mlf mkphones0.led words.mlf 명령실행

- HLEd.exe 가 실행되어 mkphones0.led와 words.mlf를 이용하여 모든 단어가 음소기호로 전환된 phones0.mlf 전사 파일 작성해줌.

```
------------------- phones0.mlf --------------------------
#!MLF!#
"*/s0001.lab"
sil
nc
uxc
rc
...
oxc
ngc
kc
.
"*/s0002.lab"
.....
------------------------------------------------------------
```

```
EX
IS sil sil
DE sp
```



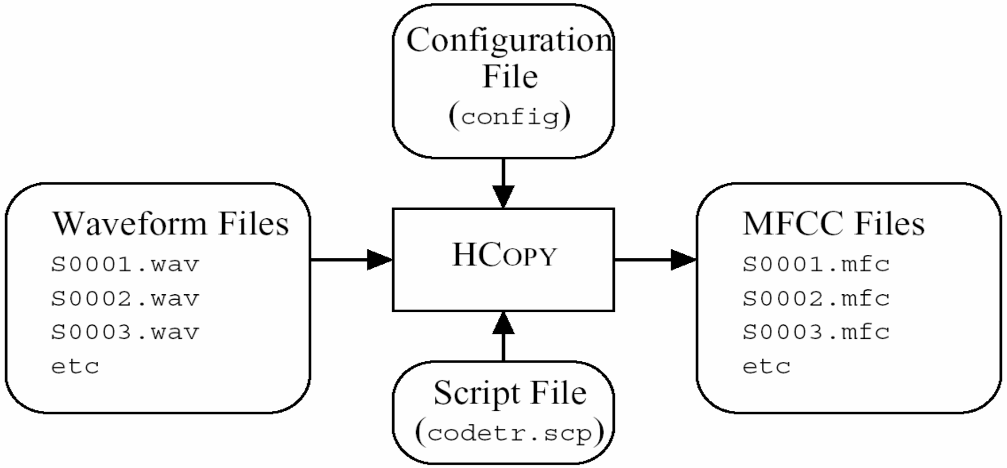
9> config 파일의 작성

-config 파일은 음성데이터를 mfc데이터로 전환시킬 때 사용되는 각 옵션들의 집합이다.

```
-------------- config ---------------------
TARGETKIND = MFCC_0
TARGETRATE = 100000.0
SOURCEFORMAT = NOHEAD
SOURCERATE = 1250
WINDOWSIZE = 250000.0
......
-------------------------------------------
```

10> codetr.scp 파일의 작성

-녹음한 음성파일명과 그것이 변환될 *.mfc파일명을 병렬적으로 적어 놓은 파일




```
------------- codetr.scp ----------------
DB\s0001.wav DB\s0001.mfc
DB\s0002.wav DB\s0002.mfc
...
DB\s0010.wav DB\s0010.mfc
......
-----------------------------------------
```

11> HCopy -T 1 -C config -S codetr.scp 명령 실행

- HCopy.exe이 config 와 codetr.scp를 이용하여 음성파일을 mfc파일로 변환시켜 줌. mfc파일은 각 음성에서 config옵션에 따라 특징값을 추출한 데이터임.

12> proto 파일과 train.scp파일의 작성

- proto 파일은 HMM 훈련에서 모델 토폴로지를 정의하는 것이다.

음소기반 시스템을 위한 3상태 left-right의 정의

```
------------------------------ proto ---------------------------
~o <vecsize> 39 <MFCC_0_D_A>
~h "proto"
<BeginHMM>
<NumStates> 5
<State>2
<Mean> 39
0.0 0.0 ....
<Variance> 39
...
...
<TransP> 5
....
<EndHMM>
-----------------------------------------------------------------
```

train.scp: 생성된 mfc파일 리스트를 포함하는 파일임

13> config1 파일의 생성

- HMM훈련을 위해 config파일의 옵션 MFCC_0 ➔ MFCC_0_D_A로 변환한 config1을 생성한다.

14> HCompV -C config1 -f 0.01 -m -S train.scp -m hmm0 proto

- HCompV.exe가 hmm0폴더에 proto파일과 vFloors파일을 생성해 준다.
이것들을 이용하여 macros 와 hmmdefs파일을 생성한다.

proto파일에 각 음소들을 포함시켜 hmmdefs파일을 생성한다.

```
-------------------- hmmdefs -------------------
~h "axc"
<BeginHMM>
...
<EndHMM>
~h "chc"
<BeginHMM>
...
<EndHMM>
...
...
-----------------------------------------------
```

vFloors 파일에 ~o를 추가하여 macros파일을 생성한다.

----------------- macros -----------------------

~o

<VecSize> 39

<MFCC_0_D_A>

~v "varFoorl"

<Variance> 39

...

Proto 파일의 일부

```
~o <VecSize> 39 <MFCC_0_D_A>
~h "proto"
```

Hmm0/vFloors

```
<Variance> 39
  1.0 1.0 1.0 ...
```

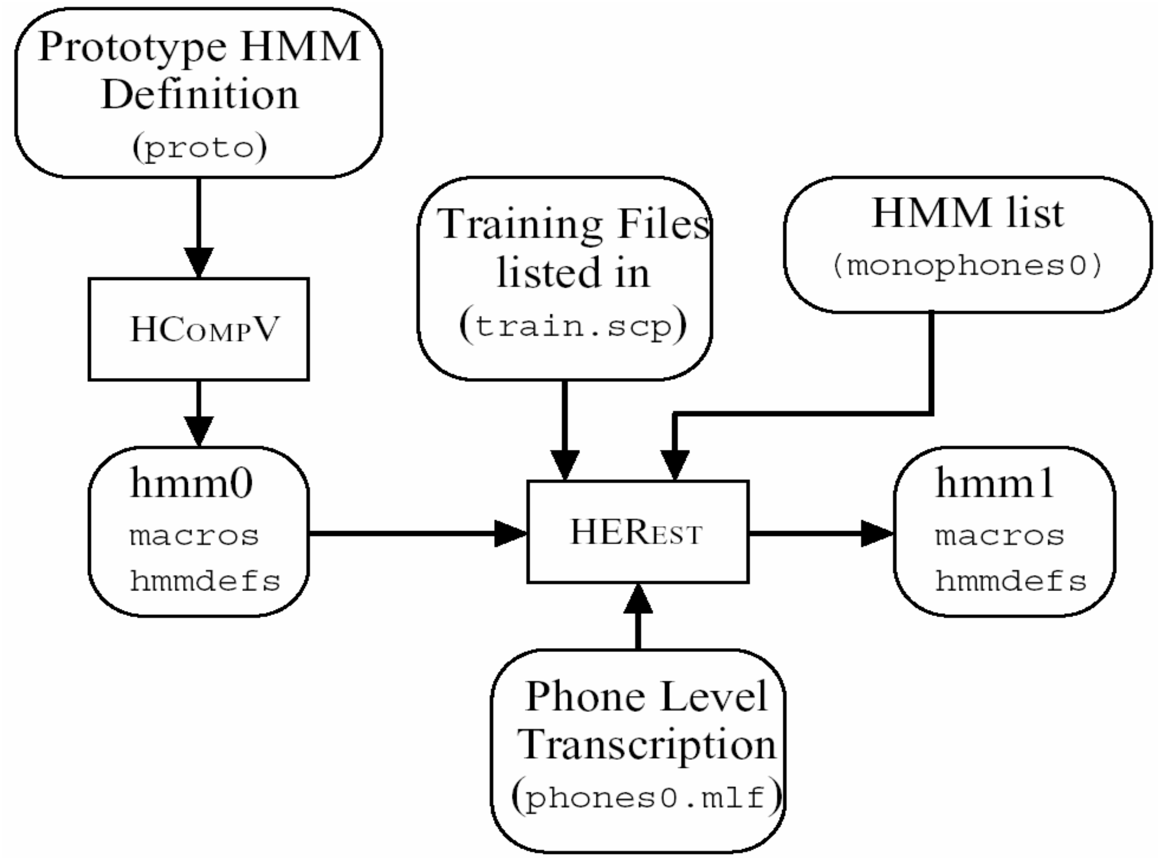
macros

```
~o
   <VecSize> 39
   <MFCC_0_D_A>
~v "varFloor1"
   <Variance> 39
      0.0012 0.0003 ...
```

15> HERest -C config1 -I phones0.mlf -t 250.0 150.0 1000.0 -S train.scp

- H hmm0\macros -H hmm0\hmmdefs -M hmm1 monophones0 명령 실행
- HERest.exe이 hmm1 폴더에 macros 와 hmmdefs 파일을 생성해준다.
- HERest.exe를 2번 실행하여 hmm2 폴더에 macros와 hmmdefs파일을 만든다.
- hmm3, hmm4, … 에 대해 반복

16> HVite -H hmm7/macros –H hmm7/hmmdefs -S test.scp -l '*' -i recout.mlf -w wdnet -p 0.0 -s 5.0 dict monophones


Prototype HMM Definition
(proto)
HCompV
Training Files listed in
(train.scp)
HMM list
(monophones0)
hmm0
macros
hmmdefs
HERest
hmm1
macros
hmmdefs
Phone Level Transcription
(phones0.mlf)

Summary

- Markov model
  - 1-st order Markov assumption on state transition
  - 'Visible': observation sequence determines state transition seq.
- Hidden Markov model
  - 1-st order Markov assumption on state transition
  - 'Hidden': observation sequence may result from many possible state transition sequences
  - Fit very well to the modeling of spatial-temporally variable signal
  - Three algorithms: model evaluation, the most probable path decoding, model training
- HMM applications and Software
  - Handwriting and speech applications
  - HMM tool box for Matlab
  - HTK
- Acknowledgement
  - 본 HMM 튜토리얼 자료를 만드는데, 상당 부분 이전 튜토리얼 자료의 사용을 허락해주신 부경대학교 신봉기 교수님과 삼성종합기술원 조성정 박사님께 감사를 표합니다.

References

- Hidden Markov Model
  - L.R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”, *IEEE Proc*. pp. 267-295, 1989.
  - L.R. Bahl et. al, “A Maximum Likelihood Approach to Continuous Speech Recognition”, *IEEE PAMI*, pp. 179-190, May. 1983.
  - M. Ostendorf, “From HMM’s to Segment Models: a Unified View of Stochastic Modeling for Speech Recognition”, *IEEE SPA*, pp 360-378, Sep., 1996.
- HMM Tutorials
  - 신봉기, “HMM Theory and Applications”, *2003컴퓨터비젼및패턴인식연구회 춘계워크샵 튜토리얼.*
  - 조성정, 한국정보과학회 ILVB Tutorial, 2005.04.16, 서울.
  - Sam Roweis, “Hidden Markov Models (SCIA Tutorial 2003)”, *http://www.cs.toronto.edu/~roweis/notes/scia03h.pdf*
  - Andrew Moore, “Hidden Markov Models”,

    http://www-2.cs.cmu.edu/~awm/tutorials/hmm.html

References (Cont.)

- HMM Applications
  - B.-K. Sin, J.-Y. Ha, S.-C. Oh, Jin H. Kim, “Network-Based Approach to Online Cursive Script Recognition”, *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics,* Vol. 29, No. 2, pp.321-328, 1999.
  - J.-Y. Ha, "Structure Code for HMM Network-Based Hangul Recognition", *18th International Conference on Computer Processing of Oriental Languages*, pp.165-170, 1999.
  - 김무중, 김효숙, 김선주, 김병기, 하진영, 권철홍, “한국인을 위한 영어 발음 교정 시스템의 개발 및 성능 평가”, *말소리*, 제46호, pp.87-102, 2003.

- HMM Topology optimization
  - H. Singer, and M. Ostendorf, “Maximum likelihood successive state splitting,” ICASSP , 1996, pp. 601-604.
  - A. Stolcke, and S. Omohundro, “Hidden Markov model induction by Bayesian model merging,” Advances in NIPS. 1993, pp. 11-18. San Mateo, CA: Morgan Kaufmann.
  - 0 A. Biem, J.-Y. Ha, J. Subrahmonia, "A Bayesian Model Selection Criterion for HMM Topology Optimization", International Conference on Acoustics Speech and Signal Processing, pp.I989~I992, IEEE Signal Processing Society, 2002.
  - A. Biem, “A Model Selection Criterion for Classification: Application to HMM Topology Optimization,” ICDAR 2003, pp. 204-210, 2003.

- HMM Software
  - Kevin Murphy, “HMM toolbox for Matlab”, freely downloadable SW written in Matlab, http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html
  - Speech Vision and Robotics Group of Cambridge University, “HTK (Hidden Markov toolkit)”, freely downloadable SW written in C, http://htk.eng.cam.ac.uk/
  - Sphinx at CMU
    http://cmusphinx.sourceforge.net/html/cmusphinx.php