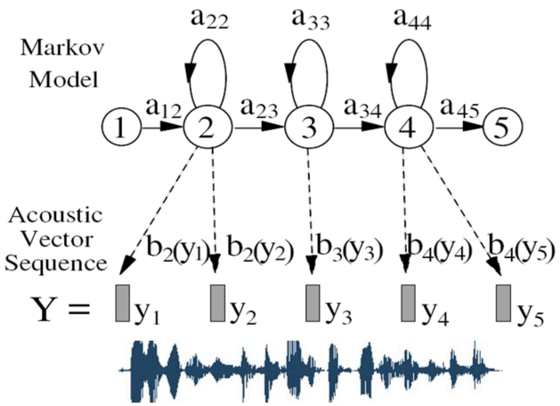


Hidden Markov Models for Speech Recognition

Berlin Chen
Department of Computer Science & Information Engineering
National Taiwan Normal University

References:

1. Rabiner and Juang. *Fundamentals of Speech Recognition*. Chapter 6
2. Huang et. al. *Spoken Language Processing*. Chapters 4, 8
3. Rabiner. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. Proceedings of the IEEE, vol. 77, No. 2, February 1989
4. Gales and Young. *The Application of Hidden Markov Models in Speech Recognition*, Chapters 1-2, 2008
5. Young. *HMMs and Related Speech Recognition Technologies*. Chapter 27, Springer Handbook of Speech Processing, Springer, 2007
6. J.A. Bilmes , *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models*, U.C. Berkeley TR-97-021

Hidden Markov Model (HMM): A Brief Overview

History

- Published in papers of Baum in late 1960s and early 1970s
- Introduced to speech processing by Baker (CMU) and Jelinek (IBM) in the 1970s (discrete HMMs)
- Then extended to continuous HMMs by Bell Labs

Assumptions

- Speech signal can be characterized as a parametric random (stochastic) process
- Parameters can be estimated in a precise, well-defined manner

Three fundamental problems

- Evaluation of probability (likelihood) of a sequence of observations given a specific HMM
- Determination of a best sequence of model states
- Adjustment of model parameters so as to best account for observed signals (or discrimination purposes)

Stochastic Process

- A stochastic process is a mathematical model of a probabilistic experiment that evolves in time and generates a sequence of numeric values
  - Each numeric value in the sequence is modeled by a random variable
  - A stochastic process is just a (finite/infinite) sequence of random variables
- Examples

  (a) The sequence of recorded values of a speech utterance

  (b) The sequence of daily prices of a stock

  (c) The sequence of hourly traffic loads at a node of a communication network

  (d) The sequence of radar measurements of the position of an airplane

Observable Markov Model

- Observable Markov Model (Markov Chain)
  - **First-order** Markov chain of *N* states is a triple (***S***,***A***,π)
    - ***S*** is a set of *N* states
    - ***A*** is the *N*×*N* matrix of transition probabilities between states
      $P(s_t=j|s_{t-1}=i, s_{t-2}=k, \ldots\ldots) \approx P(s_t=j|s_{t-1}=i) \approx A_{ij}$ First-order and time-invariant assumptions
    - π is the vector of initial state probabilities
      $\pi_j = P(s_1=j)$
  - The output of the process is the set of states at each instant of time, when each state corresponds to an observable event
  - The output in any given state is not random (***deterministic!***)
  - Too simple to describe the speech signal characteristics

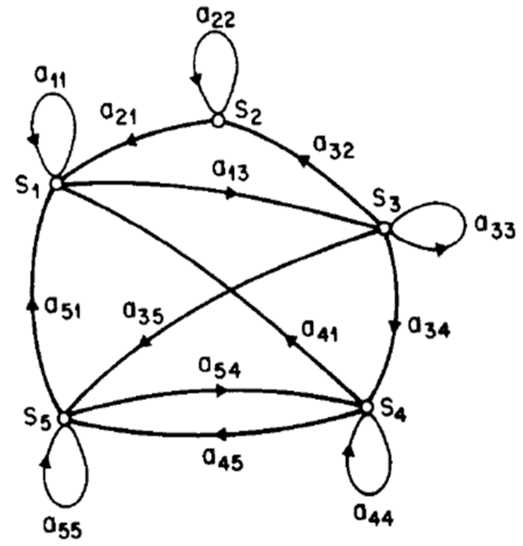
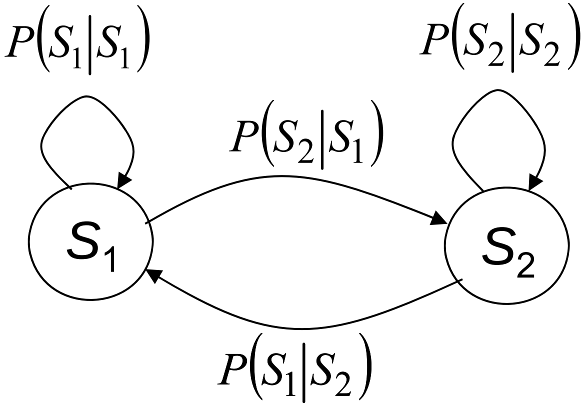
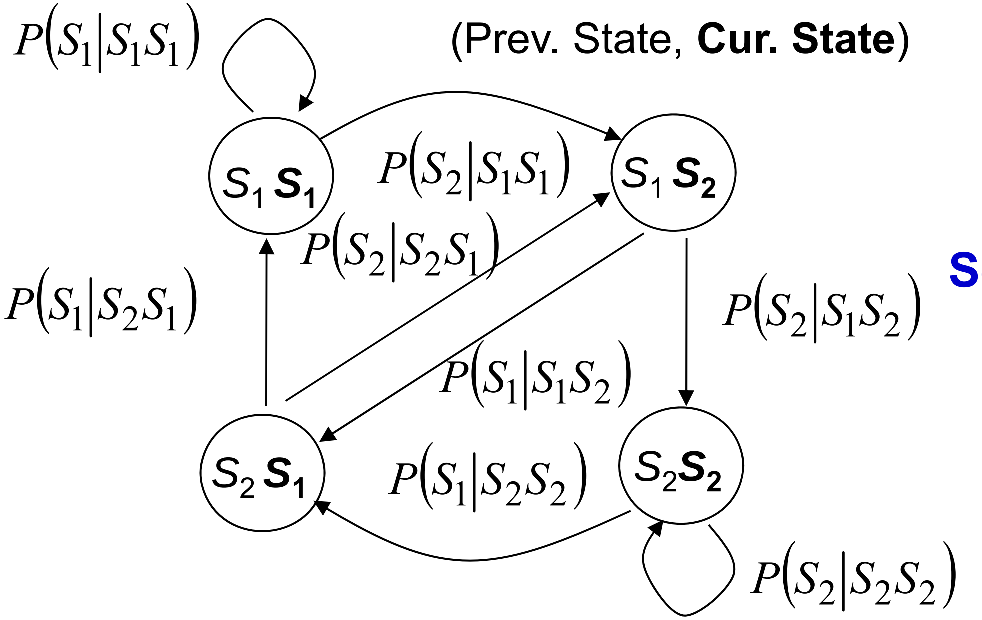


Fig. 1. A Markov chain with 5 states (labeled S_1 to S_5) with selected state transitions.

Observable Markov Model (cont.)



First-order Markov chain of *2* states



Second-order Markov chain of *2* states

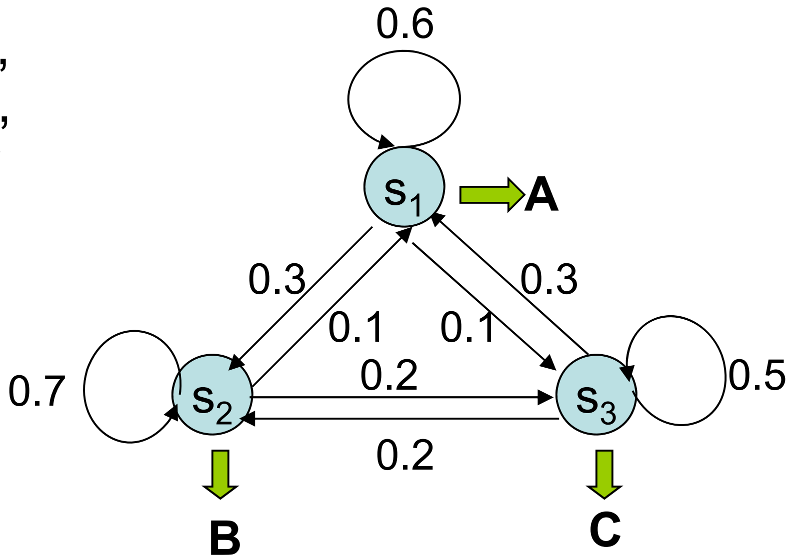
Observable Markov Model (cont.)

- Example 1: A 3-state Markov Chain λ

  State 1 generates symbol A **only**,
  State 2 generates symbol B **only**,
  and State 3 generates symbol C **only**

$$\mathbf{A}=\begin{bmatrix}0.6 & 0.3 & 0.1\\0.1 & 0.7 & 0.2\\0.3 & 0.2 & 0.5\end{bmatrix}$$

$$\pi=\begin{bmatrix}0.4 & 0.5 & 0.1\end{bmatrix}$$



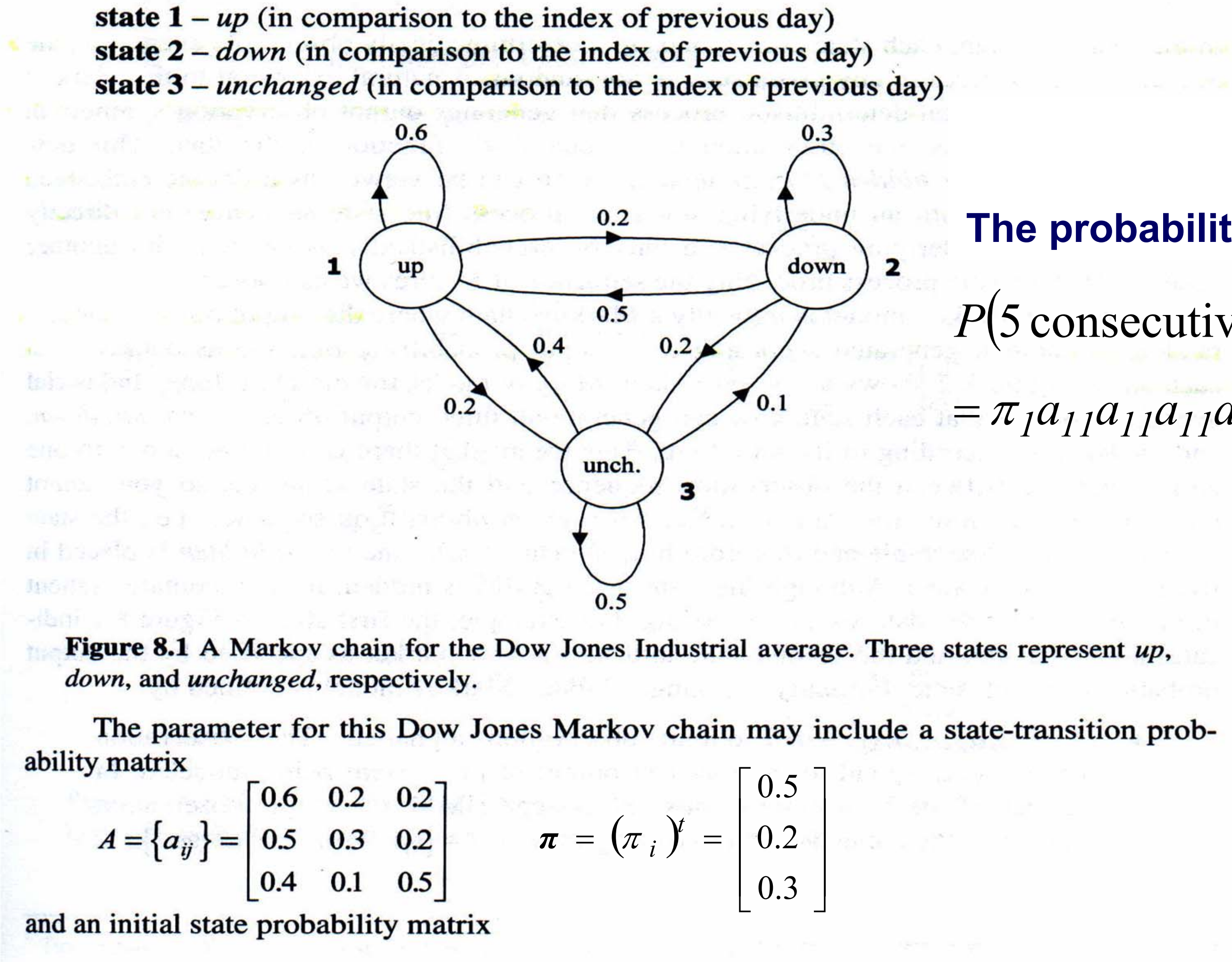
  - Given a sequence of observed symbols ***O***={CABBCABC}, the **only one** corresponding state sequence is $\{S_3S_1S_2S_2S_3S_1S_2S_3\}$, and the corresponding probability is

    $P(\boldsymbol{O}|\lambda)$

    $=P(S_3)P(S_1|S_3)P(S_2|S_1)P(S_2|S_2)P(S_3|S_2)P(S_1|S_3)P(S_2|S_1)P(S_3|S_2)$
    $=0.1\times0.3\times0.3\times0.7\times0.2\times0.3\times0.3\times0.2=0.00002268$

Observable Markov Model (cont.)

- Example 2: A three-state Markov chain for the *Dow Jones Industrial average*



state 1 – *up* (in comparison to the index of previous day)
state 2 – *down* (in comparison to the index of previous day)
state 3 – *unchanged* (in comparison to the index of previous day)

Figure 8.1 A Markov chain for the Dow Jones Industrial average. Three states represent *up*, *down*, and *unchanged*, respectively.

The parameter for this Dow Jones Markov chain may include a state-transition probability matrix

$$A = \{a_{ij}\} = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.5 & 0.3 & 0.2 \\ 0.4 & 0.1 & 0.5 \end{bmatrix}$$

and an initial state probability matrix

$$\boldsymbol{\pi} = (\pi_i)^t = \begin{bmatrix} 0.5 \\ 0.2 \\ 0.3 \end{bmatrix}$$

The probability of 5 consecutive *up* days

$$P(5 \text{ consecutive } up \text{ days}) = P(1,1,1,1,1)$$

$$= \pi_1 a_{11} a_{11} a_{11} a_{11} = 0.5 \times (0.6)^4 = 0.0648$$

Observable Markov Model (cont.)

- Example 3: Given a Markov model, what is the mean occupancy duration of each state *i*

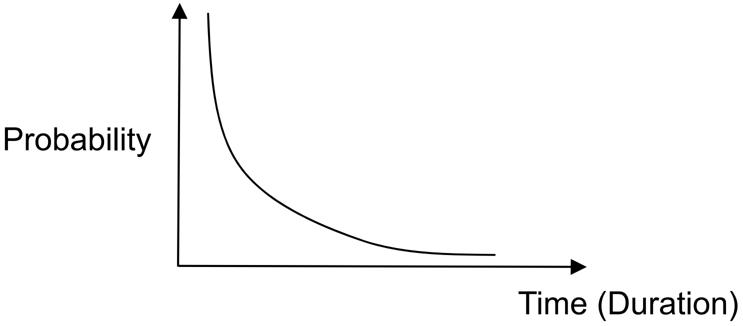
$$P_i(d) = \text{probability mass function of duration } d \text{ in state } i$$

$$= (a_{ii})^{d-1}(1-a_{ii}) \qquad \text{a geometric distribution}$$

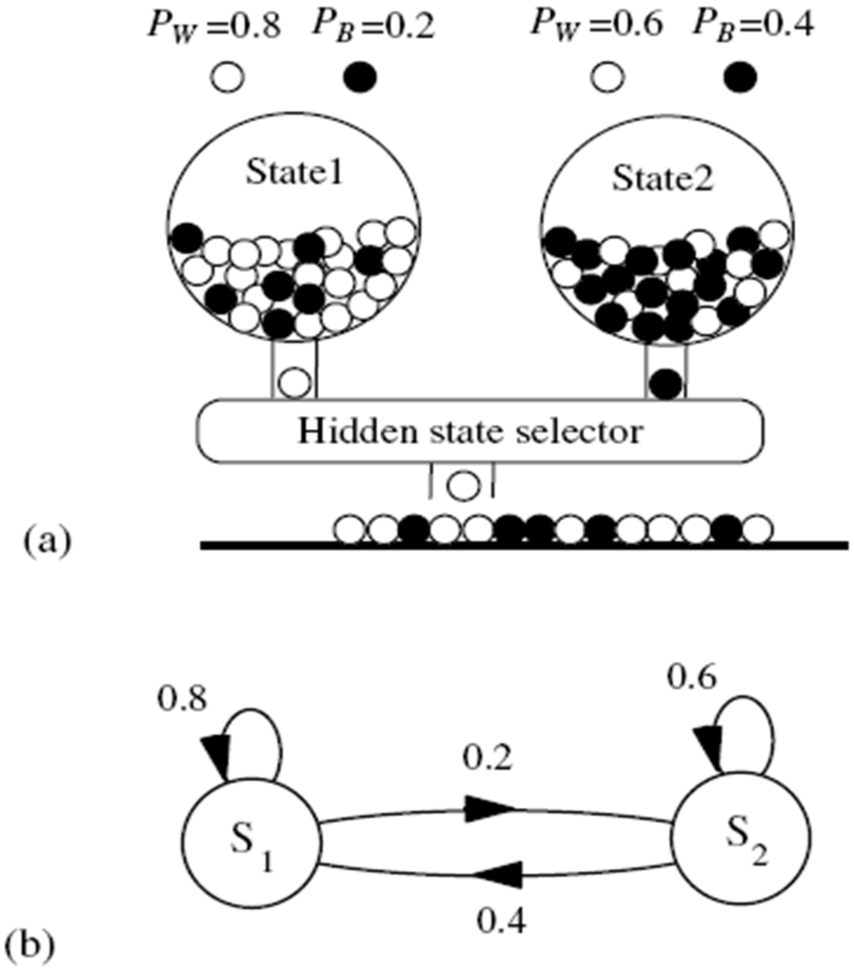
Expected number of duration in a state

$$\bar{d}_i = \sum_{d=1}^{\infty} dP_i(d) = \sum_{d=1}^{\infty} d(a_{ii})^{d-1}(1-a_{ii}) = (1-a_{ii})\frac{\partial}{\partial a_{ii}}\sum_{d=1}^{\infty}(a_{ii})^d$$

$$= (1-a_{ii})\frac{\partial}{\partial a_{ii}}\frac{1}{1-a_{ii}} = \frac{1}{1-a_{ii}}$$



Hidden Markov Model



(a) Illustration of a two-layered random process. (b) An HMM model of the process in (a).

Hidden Markov Model (cont.)

- HMM, an extended version of Observable Markov Model
  - The observation is turned to be a probabilistic function (discrete or continuous) of a state instead of an one-to-one correspondence of a state
  - The model is a doubly embedded stochastic process with an underlying stochastic process that is not directly observable (hidden)
    - What is hidden? ***The State Sequence!***
      According to the observation sequence, we are not sure which state sequence generates it!

- Elements of an HMM (the State-Output HMM) $\lambda=\{\boldsymbol{S},\boldsymbol{A},\boldsymbol{B},\pi\}$
  - ***S*** is a set of *N* states
  - ***A*** is the $N \times N$ matrix of transition probabilities between states
  - ***B*** is a set of *N* probability functions, each describing the observation probability with respect to a state
  - π is the vector of initial state probabilities

Hidden Markov Model (cont.)

- Two major assumptions
  - First order (Markov) assumption
    - The state transition depends only on the origin and destination
    - Time-invariant

$$P\left(s_t = j \middle| s_{t-1} = i\right) \approx P\left(s_\tau = j \middle| s_{\tau-1} = i\right) = P\left(j \middle| i\right) = A_{i,j}$$

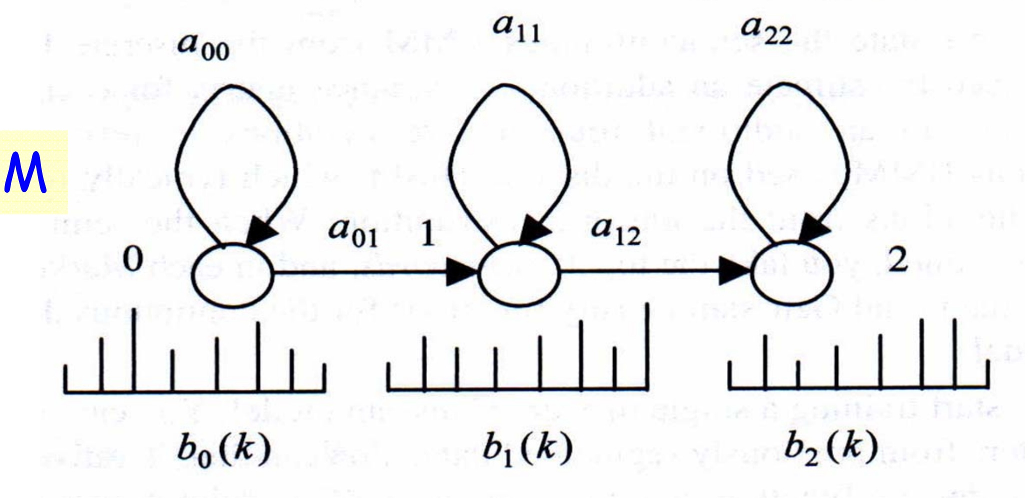
  - Output-independent assumption
    - All observations are dependent on the state that generated them, not on neighboring observations

$$P\left(\mathbf{o}_t \middle| s_t, \ldots, \mathbf{o}_{t-2}, \mathbf{o}_{t-1}, \mathbf{o}_{t+1}, \mathbf{o}_{t+2} \ldots\right) = P\left(\mathbf{o}_t \middle| s_t\right)$$

Hidden Markov Model (cont.)

- Two major types of HMMs according to the observations
  - **Discrete and finite observations:**
    - The observations that all distinct states generate are finite in number
      $V=\{v_1, v_2, v_3, \ldots\ldots, v_M\}, v_k \in R^L$
    - In this case, the set of observation probability distributions $B=\{b_j(v_k)\}$, is defined as $b_j(v_k)=P(o_t=v_k|s_t=j), 1\leq k\leq M, 1\leq j\leq N$
      o_t : *observation at time t*, s_t : *state at time t*
      ⇨ *for state j*, $b_j(v_k)$ consists of *only M probability values*

A left-to-right HMM



Hidden Markov Model (cont.)

- Two major types of HMMs according to the observations
  - **Continuous and infinite observations:**
    - The observations that all distinct states generate are infinite and continuous, that is, $\boldsymbol{V}=\{\mathbf{v} | \; \mathbf{v} \in \boldsymbol{R}^{d}\}$
    - In this case, the set of observation probability distributions $B=\{b_j(\mathbf{v})\}$, is defined as $b_j(\mathbf{v})=f_{\mathbf{O}|\mathbf{S}}(\mathbf{o}_t=\mathbf{v}|s_t=j)$, $1\leq j\leq N$
      ⇨ $b_j(\mathbf{v})$ is a ***continuous probability density function (pdf)*** ***and is often a mixture of Multivariate Gaussian (Normal) Distributions***

$$b_j(\mathbf{v})=\sum_{k=1}^{M} w_{jk}\left(\frac{1}{(2\pi)^{d/2}\left|\boldsymbol{\Sigma}_{jk}\right|^{1/2}}\exp\left(-\frac{1}{2}\left(\mathbf{v}-\boldsymbol{\mu}_{jk}\right)^{t}\boldsymbol{\Sigma}_{jk}^{-1}\left(\mathbf{v}-\boldsymbol{\mu}_{jk}\right)\right)\right)$$

Mixture Weight

Covariance Matrix

Observation Vector

Mean Vector

Hidden Markov Model (cont.)

- Multivariate Gaussian Distributions
  - When ***X***=($\boldsymbol{x}_1$, $\boldsymbol{x}_2$,…, $\boldsymbol{x}_d$) is a *d*-dimensional random vector, the multivariate Gaussian pdf has the form:

$$f\left(\mathbf{X}=\mathbf{x}\middle|\boldsymbol{\mu},\boldsymbol{\Sigma}\right)=N\left(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}\right)=\frac{1}{\left(2\pi\right)^{d/2}\left|\boldsymbol{\Sigma}\right|^{1/2}}\exp\left(-\frac{1}{2}\left(\mathbf{x}-\boldsymbol{\mu}\right)^{t}\boldsymbol{\Sigma}^{-1}\left(\mathbf{x}-\boldsymbol{\mu}\right)\right)$$

where $\boldsymbol{\mu}$ is the L - dimensional mean vector, $\boldsymbol{\mu}=E\left[\mathbf{x}\right]$

$\boldsymbol{\Sigma}$ is the coverance matrix, $\boldsymbol{\Sigma}=E\left[\left(\mathbf{x}-\boldsymbol{\mu}\right)\left(\mathbf{x}-\boldsymbol{\mu}\right)^{t}\right]=E\left[\mathbf{x}\mathbf{x}^{t}\right]-\boldsymbol{\mu}\boldsymbol{\mu}^{t}$

and $\left|\boldsymbol{\Sigma}\right|$ is the the determinant of $\boldsymbol{\Sigma}$

The i-j^{th} elevment σ_{ij} of $\boldsymbol{\Sigma}$, $\sigma_{ij}=E\left[\left(x_i-\mu_i\right)\left(x_j-\mu_j\right)\right]=E\left[x_ix_j\right]-\mu_i\mu_j$

  - If $\boldsymbol{x}_1$, $\boldsymbol{x}_2$,…, $\boldsymbol{x}_d$ are independent, the covariance matrix is reduced to diagonal covariance
    - Viewed as *d* independent scalar Gaussian distributions
    - Model complexity is significantly reduced

Hidden Markov Model (cont.)

- Multivariate Gaussian Distributions

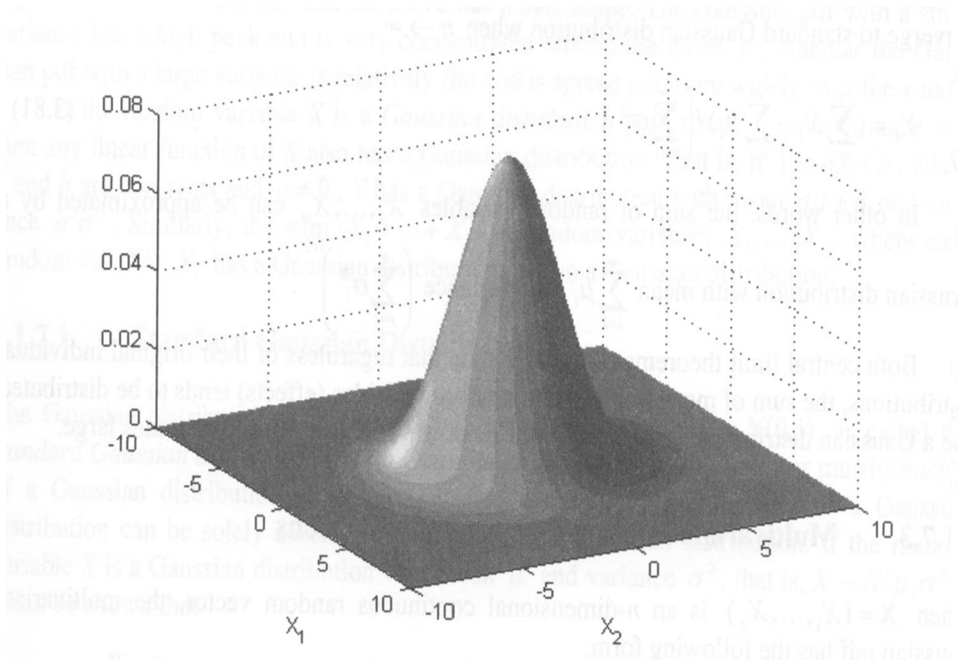


Figure 3.12 A two-dimensional multivariate Gaussian distribution with independent random variables x_1 and x_2 that have the same variance.

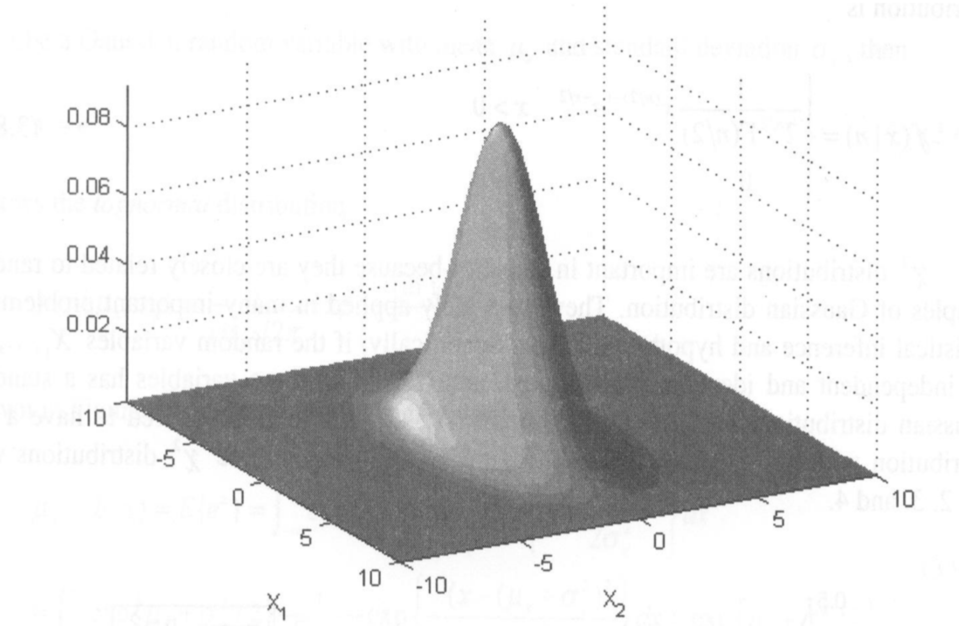
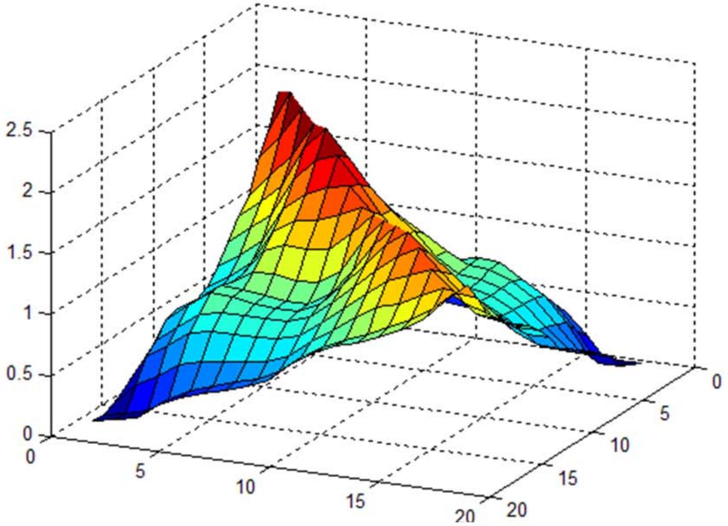


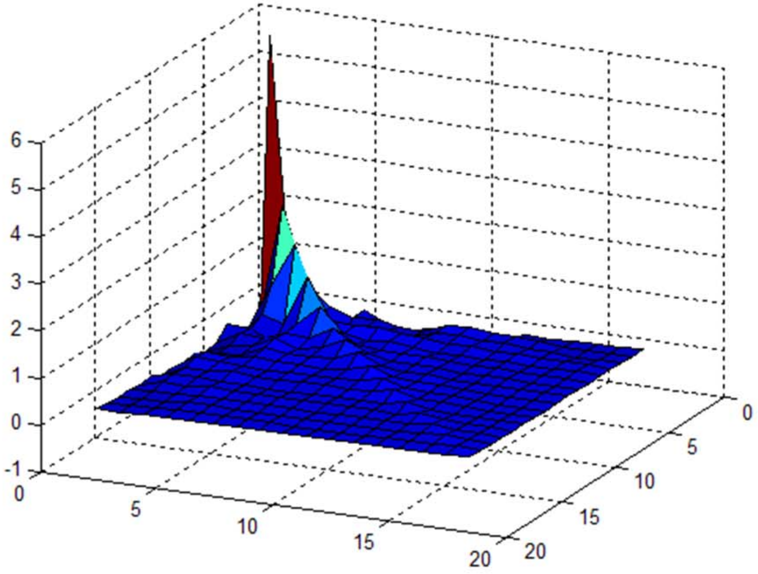
Figure 3.13 Another two-dimensional multivariate Gaussian distribution with independent random variable x_1 and x_2 which have different variances.

Hidden Markov Model (cont.)

- Covariance matrix of the correlated feature vectors (Mel-frequency filter bank outputs)



- Covariance matrix of the partially de-correlated feature vectors (MFCC without C_0)
  - MFCC: Mel-frequency cepstral coefficients

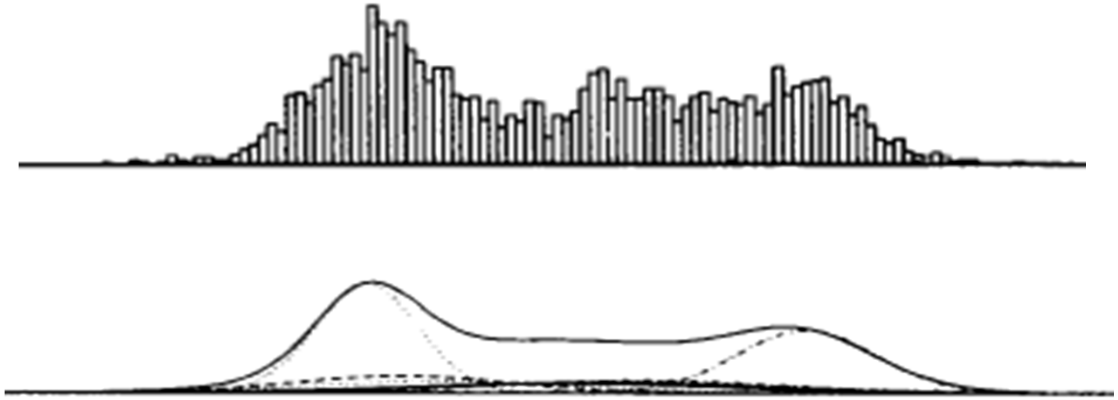


Hidden Markov Model (cont.)

- Multivariate *Mixture* Gaussian Distributions (cont.)
  - More complex distributions with multiple local maxima can be approximated by Gaussian (a unimodal distribution) mixtures

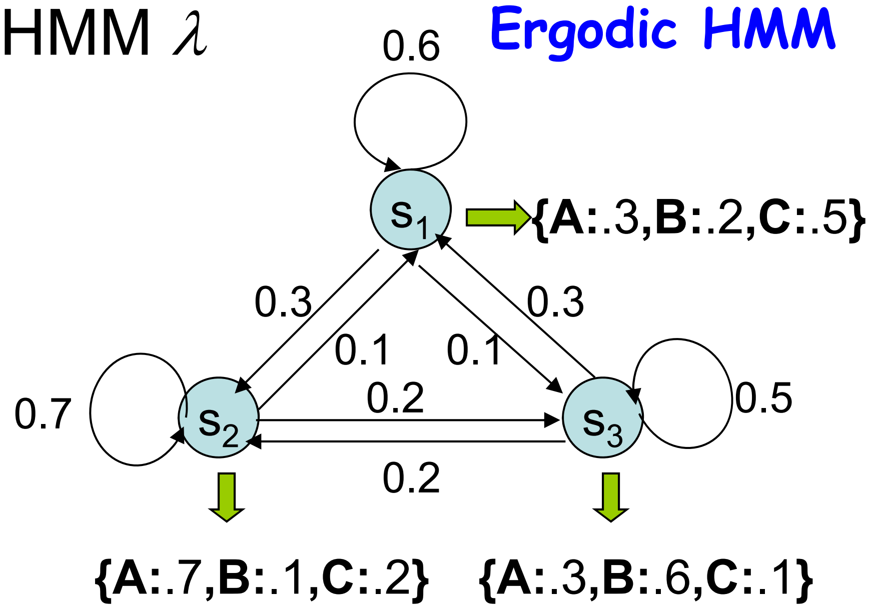
$$f(\boldsymbol{x}) = \sum_{k=1}^{M} w_k N_k(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \qquad \sum_{k=1}^{M} w_k = 1$$

  - Gaussian mixtures with enough mixture components can approximate any distribution



Hidden Markov Model (cont.)

- Example 4: a 3-state discrete HMM λ



$$\mathbf{A} = \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.1 & 0.7 & 0.2 \\ 0.3 & 0.2 & 0.5 \end{bmatrix}$$

$$b_1(\mathbf{A}) = 0.3, b_1(\mathbf{B}) = 0.2, b_1(\mathbf{C}) = 0.5$$

$$b_2(\mathbf{A}) = 0.7, b_2(\mathbf{B}) = 0.1, b_2(\mathbf{C}) = 0.2$$

$$b_3(\mathbf{A}) = 0.3, b_3(\mathbf{B}) = 0.6, b_3(\mathbf{C}) = 0.1$$

$$\pi = [0.4 \quad 0.5 \quad 0.1]$$

  - Given a sequence of observations *O*={*ABC*}, there are **27 possible** corresponding state sequences, and therefore the corresponding probability is

$$P(\boldsymbol{O}|\lambda) = \sum_{i=1}^{27} P(\boldsymbol{O}, \boldsymbol{S}_i|\lambda) = \sum_{i=1}^{27} P(\boldsymbol{O}|\boldsymbol{S}_i, \lambda) P(\boldsymbol{S}_i|\lambda), \qquad \boldsymbol{S}_i : \text{state sequence}$$

$$E.g. \text{ when } \boldsymbol{S}_i = \{s_2 s_2 s_3\}, P(\boldsymbol{O}|\boldsymbol{S}_i, \boldsymbol{\lambda}) = P(\boldsymbol{A}|s_2) P(\boldsymbol{B}|s_2) P(\boldsymbol{C}|s_3) = 0.7 * 0.1 * 0.1 = 0.007$$

$$P(\boldsymbol{S}_i|\boldsymbol{\lambda}) = P(s_2) P(s_2|s_2) P(s_3|s_2) = 0.5 * 0.7 * 0.2 = 0.07$$

Hidden Markov Model (cont.)

- Notations:
  - **$\boldsymbol{O}=\{o_1o_2o_3\ldots\ldots o_T\}$:** the observation (feature) sequence
  - **$\boldsymbol{S}=\{s_1s_2s_3\ldots\ldots s_T\}$:** the state sequence
  - λ : model, for HMM, $\lambda=\{\boldsymbol{A},\boldsymbol{B},\pi\}$
  - $P(\boldsymbol{O}|\lambda)$: The probability of observing ***O*** given the model λ
  - $P(\boldsymbol{O}|\boldsymbol{S},\lambda)$: The probability of observing ***O*** given λ and a state sequence ***S*** of λ
  - $P(\boldsymbol{O},\boldsymbol{S}|\lambda)$: The probability of observing ***O*** and ***S*** given λ
  - $P(\boldsymbol{S}|\boldsymbol{O},\lambda)$: The probability of observing ***S*** given ***O*** and λ
- Useful formulas
  - Bayes' Rule :

$$P(A|B)=\frac{P(A,B)}{P(B)}=\frac{P(B|A)P(A)}{P(B)} \Longrightarrow P(A|B,\lambda)=\frac{P(A,B|\lambda)}{P(B|\lambda)}=\frac{P(B|A,\lambda)P(A|\lambda)}{P(B|\lambda)}$$

λ : model describing the probability

$$P(A,B)=P(B|A)P(A)=P(A|B)P(B)$$

chain rule

Hidden Markov Model (cont.)

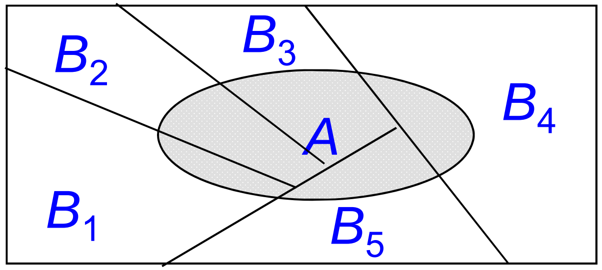
- Useful formulas (Cont.):
  - Total Probability Theorem

marginal probability

$$P(A)=\begin{cases}\sum_{\text{all } B}P(A,B)=\sum_{\text{all } B}P(A|B)P(B), & \text{if } B \text{ is disrete and disjoint}\\ \int_B f(A,B)dB=\int_B f(A|B)f(B)dB, & \text{if } B \text{ is continuous}\end{cases}$$

if $x_1,x_2,\ldots\ldots,x_n$ are independent,

$$\Rightarrow P(x_1,x_2,\ldots\ldots,x_n)=P(x_1)P(x_2)\ldots\ldots P(x_n)$$



Venn Diagram

$$E_z[q(z)]=\begin{cases}\sum_k P(z=k)q(k), & z:\text{discrete}\\ \int_z f_{\mathbf{z}}(z)q(z)dz, & z:\text{continuous}\end{cases}$$

Expectation

Three Basic Problems for HMM

- Given an observation sequence $O=(o_1,o_2,.....,o_T)$, and an HMM $\lambda=(S,A,B,\pi)$
  - **Problem *1***:

    How to *efficiently* compute $P(\mathbf{O}|\lambda)$?

    ⇨ ***Evaluation problem***
  - **Problem *2*:**

    How to choose an optimal state sequence $\mathbf{S}=(s_1,s_2,......, s_T)$?

    ⇨ ***Decoding Problem***
  - **Problem *3*:**

    How to adjust the model parameter $\boldsymbol{\lambda}=(\mathbf{A},B,\pi)$ to maximize $P(\mathbf{O}|\lambda)$?

    ⇨ ***Learning / Training Problem***

Basic Problem 1 of HMM (cont.)

Given ***O*** and λ, find $P(\boldsymbol{O}|\lambda)$= Prob[observing *O* given λ]

- Direct Evaluation
  - Evaluating all possible state sequences of length *T* that generating observation sequence ***O***

$$P(\boldsymbol{O}|\lambda) = \sum_{all\ \boldsymbol{S}} P(\boldsymbol{O}, \boldsymbol{S}|\lambda) = \sum_{all\ S} P(\boldsymbol{O}|\boldsymbol{S}, \lambda) P(\boldsymbol{S}|\lambda)$$

  - $P(\boldsymbol{S}|\lambda)$: The probability of each path ***S***
    - By Markov assumption (First-order HMM)

$$P(\boldsymbol{S}|\lambda) = P(s_1|\lambda) \prod_{t=2}^{T} P\left(s_t \middle| s_1^{t-1}, \lambda\right)$$

By chain rule

$$\approx P(s_1|\lambda) \prod_{t=2}^{T} P(s_t|s_{t-1}, \lambda)$$

By Markov assumption

$$= \pi_{s_1} a_{s_1 s_2} a_{s_2 s_3} \ldots a_{s_{T-1} s_T}$$

Basic Problem 1 of HMM (cont.)

- Direct Evaluation (cont.)
  - $P(\boldsymbol{O}|\boldsymbol{S},\lambda)$: The joint output probability along the path *S*
    - By output-independent assumption
      - The probability that a particular observation symbol/vector is emitted at time *t* depends only on the state s_t and is conditionally independent of the past observations

$$
\begin{aligned}
P(\boldsymbol{O}|\boldsymbol{S},\lambda) &= P\left(\boldsymbol{o}_1^T \middle| s_1^T, \lambda\right) \\
&= P\left(\boldsymbol{o}_1 \middle| s_1^T, \lambda\right) \prod_{t=2}^{T} P\left(\boldsymbol{o}_t \middle| \boldsymbol{o}_1^{t-1}, s_1^T, \lambda\right) \\
&\approx \prod_{t=1}^{T} P\left(\boldsymbol{o}_t \middle| s_t, \lambda\right) \quad \text{By output-independent assumption} \\
&= \prod_{t=1}^{T} b_{s_t}(\boldsymbol{o}_t)
\end{aligned}
$$

Basic Problem 1 of HMM (cont.)

- Direct Evaluation (Cont.)

$$P(\boldsymbol{o}_t | s_t, \lambda) = b_{s_t}(\boldsymbol{o}_t)$$

$$P(\boldsymbol{O}|\lambda) = \sum_{all\ \boldsymbol{S}} P(\boldsymbol{S}|\lambda) P(\boldsymbol{O}|\boldsymbol{S}, \lambda)$$

$$= \sum_{all\ \boldsymbol{s}} \left(\left[\pi_{s_1} a_{s_1 s_2} a_{s_2 s_3} \ldots\ldots a_{s_{T-1} s_T} \right] \left[b_{s_1}(\boldsymbol{o}_1) b_{s_2}(\boldsymbol{o}_2) \ldots\ldots b_{s_T}(\boldsymbol{o}_T) \right] \right)$$

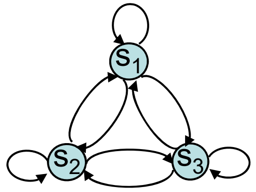
$$= \sum_{s_1, s_2, .., s_T} \pi_{s_1} b_{s_1}(\boldsymbol{o}_1) a_{s_1 s_2} b_{s_2}(\boldsymbol{o}_2) \ldots\ldots a_{s_{T-1} s_T} b_{s_T}(\boldsymbol{o}_T)$$

  - Huge Computation Requirements: $O(N^T)$
    - Exponential computational complexity

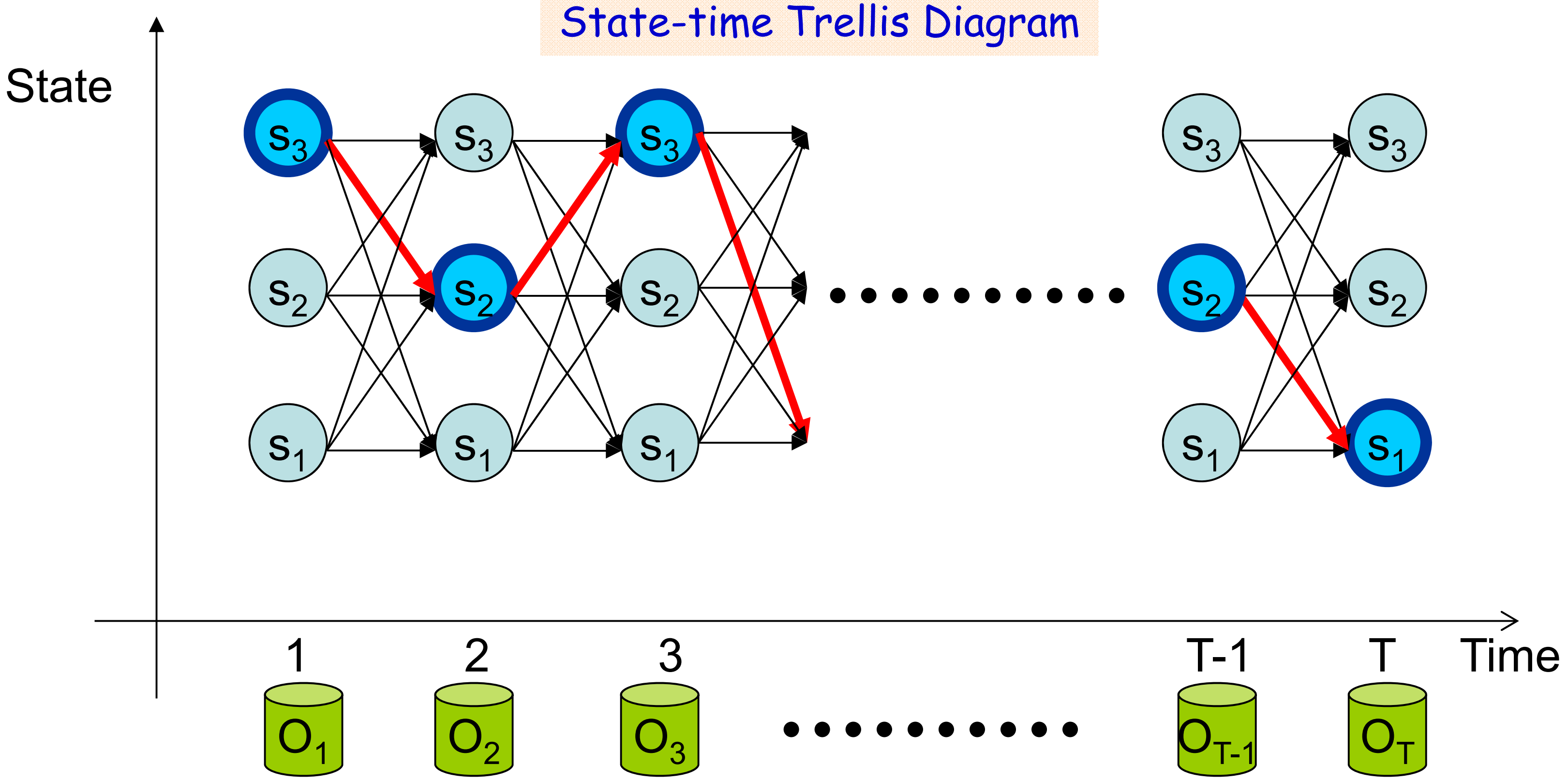
$$\textbf{\textit{Complexity}} \quad : (2T\text{-}1) N^T\ MUL \quad \approx 2TN^T,\ N^T\text{-}1\ \text{ADD}$$

- A more efficient algorithms can be used to evaluate $P(\boldsymbol{O}|\lambda)$
  - *Forward/Backward Procedure/Algorithm*

Basic Problem 1 of HMM (cont.)



- Direct Evaluation (Cont.)


State-time Trellis Diagram

s_i denotes that $b_j(o_t)$ has been computed

a_{ij} → denotes that a_{ij} has been computed

Basic Problem 1 of HMM
- The Forward Procedure

- Based on the HMM assumptions, the calculation of $P\left(s_t \middle| s_{t-1}, \lambda\right)$ and $P\left(\boldsymbol{o}_t \middle| s_t, \lambda\right)$ involves only s_{t-1} , s_t and $\boldsymbol{o}_t$, so it is possible to compute the likelihood with recursion on t

- Forward variable : $\alpha_t(i) = P\left(o_1 o_2 \ldots o_t, s_t = i \middle| \boldsymbol{\lambda}\right)$
  - The probability that the HMM is in state *i* at time *t* having generating partial observation $\boldsymbol{o}_1\boldsymbol{o}_2\ldots\boldsymbol{o}_t$

Basic Problem 1 of HMM

- The Forward Procedure (cont.)

- Algorithm

  1. Initialization $\alpha_1(i) = \pi_i b_i(\boldsymbol{o}_1),\ 1 \le i \le N$

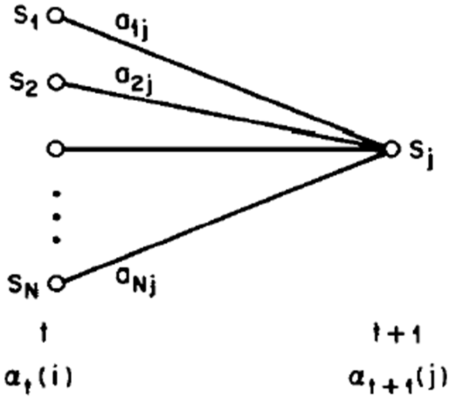
  2. Induction $\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(\boldsymbol{o}_{t+1}),\ \ 1 \le t \le T\text{-}1, 1 \le j \le N$

  3. Termination $P(\boldsymbol{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$

  – Complexity: $O(N^2T)$

    $\text{MUL} \quad : N(N+1)(T-1)+N \quad \approx \quad N^2T$

    $\text{ADD} \quad : (N-1)N(T-1)+(N-1) \approx N^2T$



- Based on the lattice (trellis) structure
  – Computed in a *time-synchronous* fashion from *left-to-right*, where each cell for time *t* is completely computed before proceeding to time *t+1*

- All state sequences, regardless how long previously, merge to *N* nodes (states) at each time instance *t*

Basic Problem 1 of HMM
- The Forward Procedure (cont.)

$$\alpha_t(j) = P(o_1 o_2 \ldots o_t, s_t = j | \lambda)$$

$$= P(o_1 o_2 \ldots o_t | s_t = j, \lambda) P(s_t = j | \lambda)$$

$$= P(o_1 o_2 \ldots o_{t-1} | s_t = j, \lambda) P(o_t | s_t = j, \lambda) P(s_t = j | \lambda)$$

$$= P(o_1 o_2 \ldots o_{t-1}, s_t = j | \lambda) P(o_t | s_t = j, \lambda)$$

$$= P(o_1 o_2 \ldots o_{t-1}, s_t = j | \lambda) b_j(o_t)$$

$$= \left[\sum_{i=1}^{N} P(o_1 o_2 \ldots o_{t-1}, s_{t-1} = i, s_t = j | \lambda) \right] b_j(o_t)$$

$$= \left[\sum_{i=1}^{N} P(o_1 o_2 \ldots o_{t-1}, s_{t-1} = i | \lambda) P(s_t = j | o_1 o_2 \ldots o_{t-1}, s_{t-1} = i, \lambda) \right] b_j(o_t)$$

$$= \left[\sum_{i=1}^{N} P(o_1 o_2 \ldots o_{t-1}, s_{t-1} = i | \lambda) P(s_t = j | s_{t-1} = i, \lambda) \right] b_j(o_t)$$

$$= \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} \right] b_j(o_t)$$

$P(A,B) = P(B|A)P(A)$

output independent assumption

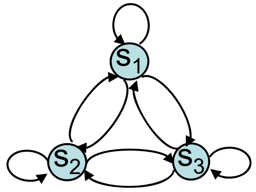
$P(B|A)P(A) = P(A,B)$

$P(o_t|s_t = j, \lambda) = b_j(o_t)$

$P(A) \sum_{all\,B} P(A,B)$

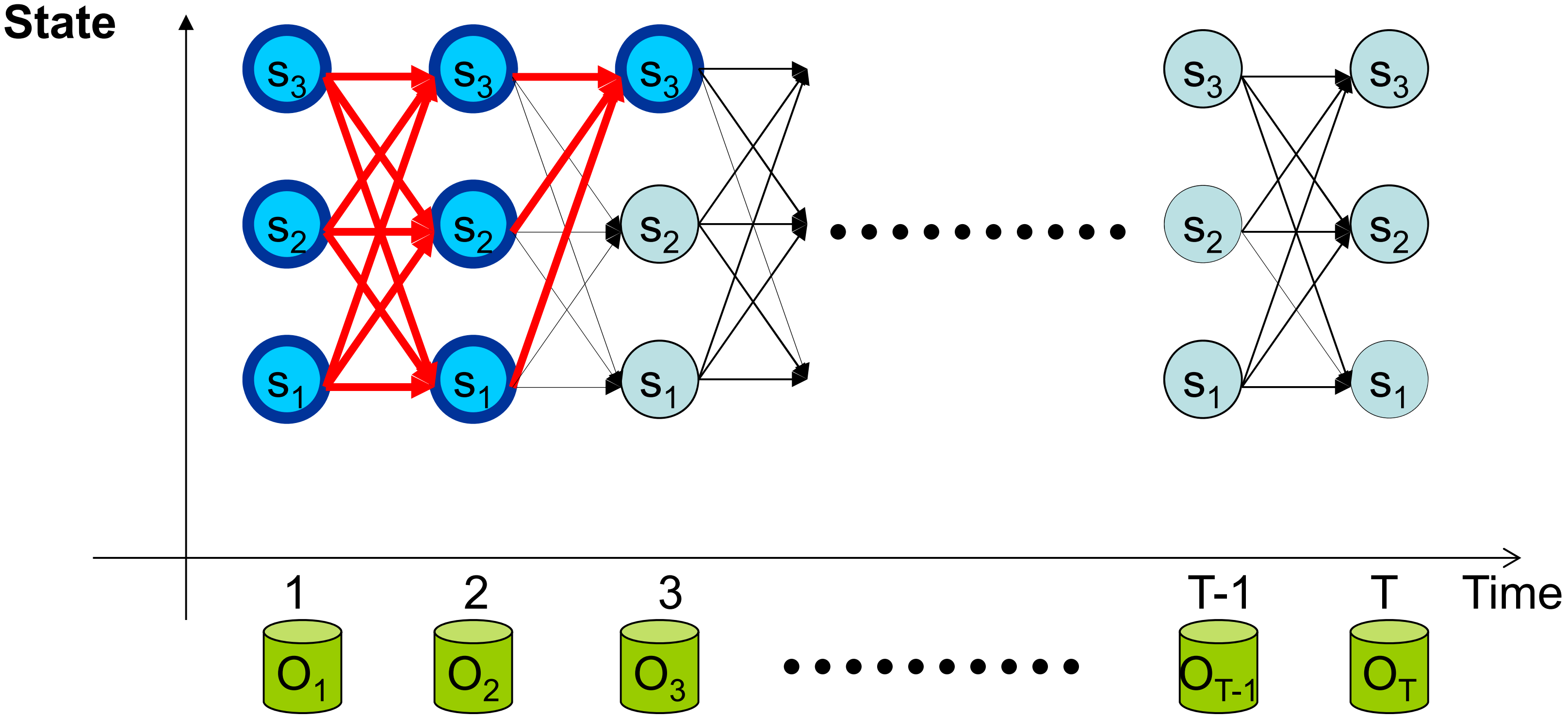
first-order Markov assumption

Basic Problem 1 of HMM
- The Forward Procedure (cont.)



- $\alpha_3(3)=P(\boldsymbol{o}_1,\boldsymbol{o}_2,\boldsymbol{o}_3,s_3=\mathbf{3}|\lambda)$

  $=[\alpha_2(1)^*a_{13}+\alpha_2(2)^*a_{23}+\alpha_2(3)^*a_{33}]b_3(\boldsymbol{o}_3)$

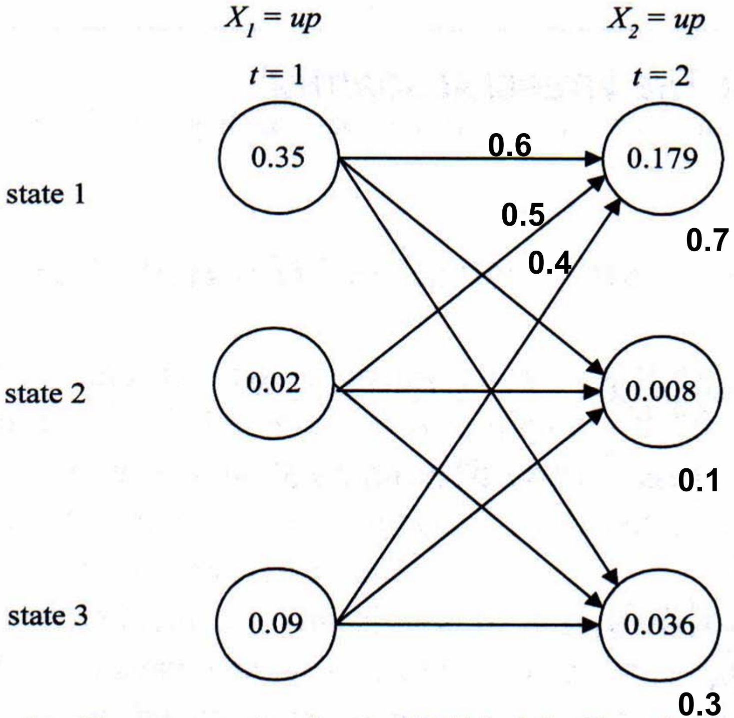


s_i **denotes that** $b_j(o_t)$ **has been computed**

a_{ij} **denotes** a_{ij} **has been computed**

Basic Problem 1 of HMM
- The Forward Procedure (cont.)

- A three-state Hidden Markov Model for the *Dow Jones Industrial average*



(0.6*0.35+0.5*0.02+0.4*0.009)*0.7
=0.1792

Figure 8.4 The forward trellis computation for the HMM of the Dow Jones Industrial average.

Basic Problem 1 of HMM
- The Backward Procedure

- Backward variable : $\beta_t(i)=P(\boldsymbol{o}_{t+1},\boldsymbol{o}_{t+2},\ldots..,\boldsymbol{o}_T|s_t=i\ ,\ \lambda)$

$$1.\ \text{Initialization}: \beta_{\mathrm{T}}(i)=1,\ 1\le i\le N$$

$$2.\ \text{Induction}: \beta_t(i)=\sum_{j=1}^{N} a_{ij} b_j(\boldsymbol{o}_{t+1})\beta_{t+1}(j),\ 1\le t\le T\text{-}1, 1\le i\le N$$

$$3.\ \text{Termination}: P(\boldsymbol{O}|\lambda)=\sum_{j=1}^{N}\pi_j b_j(\boldsymbol{o}_1)\beta_1(j)$$

$$\text{Complexity MUL}: 2N^2(T\text{-}1)+2N\approx\ N^2T\ ;$$

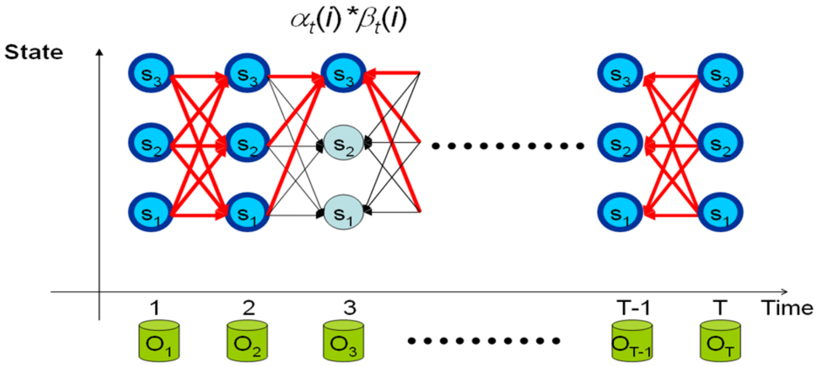
$$\text{ADD}: (N\text{-}1)N(T\text{-}1)+N\approx N^2T$$

Basic Problem 1 of HMM

- Backward Procedure (cont.)

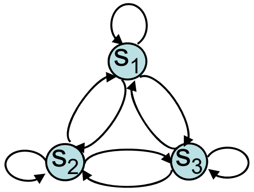
- Why $P(\boldsymbol{O}, s_t = i|\lambda) = \alpha_t(i)\beta_t(i)$?

$$\begin{aligned}
&\alpha_t(i)\beta_t(i)\\
&= P(\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_t, s_t = i|\lambda) \cdot P(\boldsymbol{o}_{t+1}, \boldsymbol{o}_{t+2}, \ldots, \boldsymbol{o}_T | s_t = i, \lambda)\\
&= P(\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_t | s_t = i, \lambda) P(s_t = i|\lambda) P(\boldsymbol{o}_{t+1}, \boldsymbol{o}_{t+2}, \ldots, \boldsymbol{o}_T | s_t = i, \lambda)\\
&= P(\boldsymbol{o}_1, \ldots, \boldsymbol{o}_t, \ldots, \boldsymbol{o}_T | s_t = i, \lambda) P(s_t = i|\lambda)\\
&= P(\boldsymbol{o}_1, \ldots, \boldsymbol{o}_t, \ldots, \boldsymbol{o}_T, s_t = i|\lambda)\\
&= P(\boldsymbol{O}, s_t = i|\lambda)
\end{aligned}$$



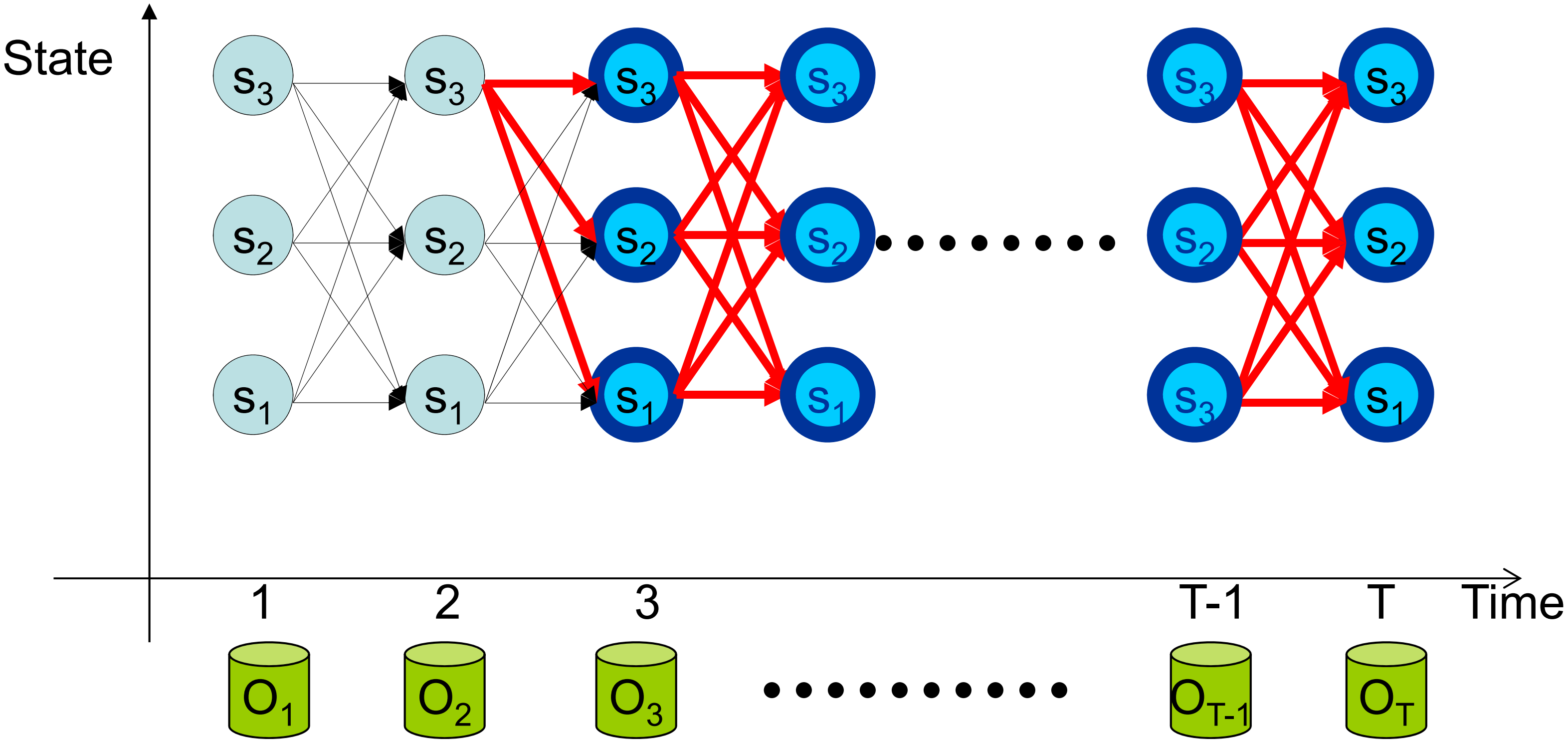
- $P(\boldsymbol{O}|\lambda) = \sum_{i=1}^{N} P(\boldsymbol{O}, s_t = i|\lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i)$

Basic Problem 1 of HMM
- The Backward Procedure (cont.)

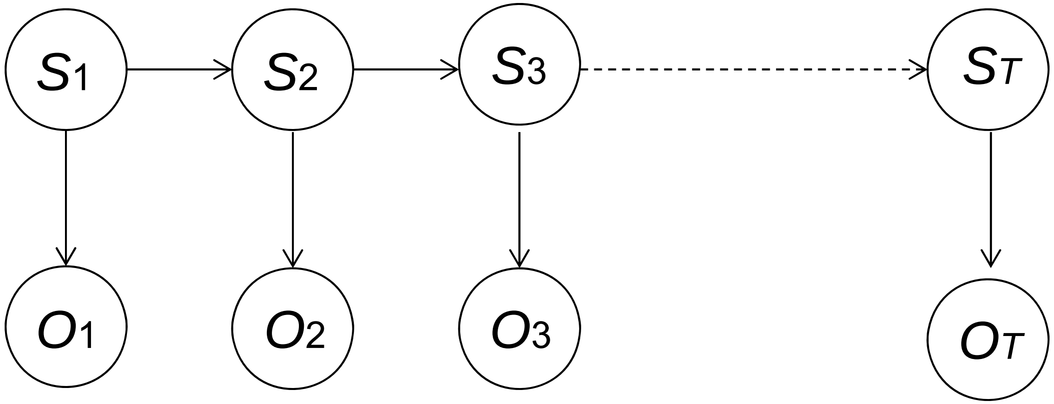


- $\beta_2(3)=P(o_{\mathbf{3}},o_{\mathbf{4}},\ldots,\ o_T|s_2\mathbf{=3,}\lambda)$

$$=a_{31}*\ b_1(o_3)*\beta_3(1)\ +a_{32}*\ b_2(o_3)*\beta_3(2)+a_{33}*\ b_1(o_3)*\beta_3(3)$$



HMM is a Kind of Bayesian Network



Basic Problem 2 of HMM

How to choose an optimal state sequence $S=(s_1,s_2,\ldots\ldots, s_T)$?

- The first optimal criterion: **Choose the states s_t are *individually* most likely at each time *t***

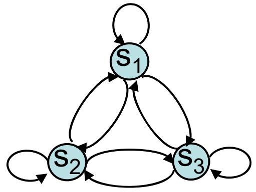
Define a posteriori probability variable $\gamma_t(i)=P(s_t=i|\boldsymbol{O},\boldsymbol{\lambda})$

$$\gamma_t(i)=\frac{P(s_t=i,\boldsymbol{O}|\boldsymbol{\lambda})}{P(\boldsymbol{O}|\boldsymbol{\lambda})}=\frac{P(s_t=i,\boldsymbol{O}|\boldsymbol{\lambda})}{\sum_{m=1}^{N}P(s_t=m,\boldsymbol{O}|\boldsymbol{\lambda})}=\frac{\alpha_t(i)\,\beta_t(i)}{\sum_{m=1}^{N}\alpha_t(m)\,\beta_t(m)}$$

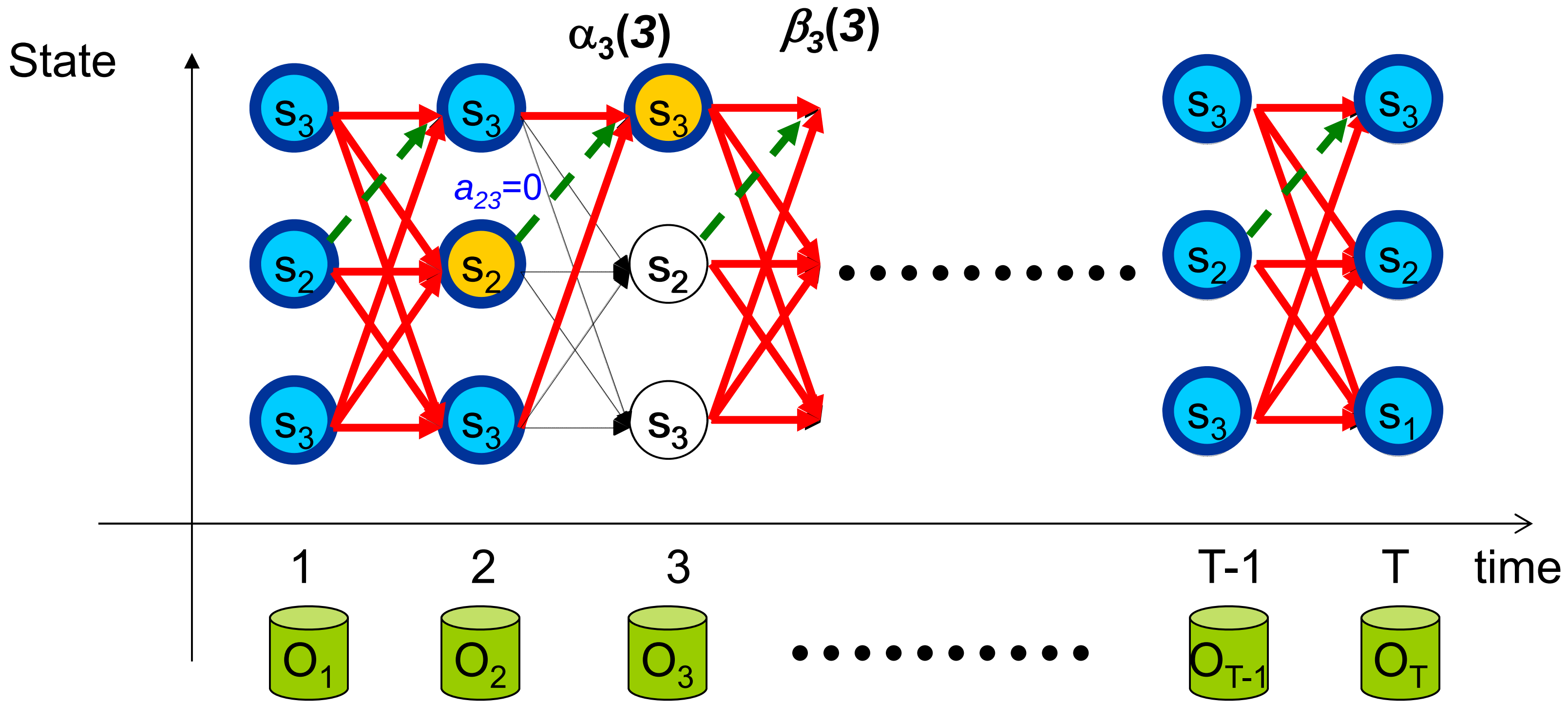
state occupation probability (count) – a soft alignment of HMM state to the observation (feature)

- Solution : $s_t^* = arg_i \max [\gamma_t(i)],\ 1 \le t \le T$
  - Problem: maximizing the probability at each time *t* individually $\boldsymbol{S}^*= s_1^* s_2^* \ldots s_T^*$ may not be a valid sequence (e.g. $a_{s_t^* s_{t+1}^*} = 0$)

Basic Problem 2 of HMM (cont.)



- $P(s_3 = 3 , \mathbf{O} \mid \lambda) = \alpha_3(3) * \beta_3(3)$



Basic Problem 2 of HMM

- The Viterbi Algorithm

- The second optimal criterion: The Viterbi algorithm can be regarded as the dynamic programming algorithm applied to the HMM or as a modified forward algorithm
  - Instead of summing up probabilities from different paths coming to the same destination state, the Viterbi algorithm picks and remembers the best path
    - Find a single optimal state sequence $\boldsymbol{S}=(s_1,s_2,\ldots\ldots, s_T)$
      - How to find the second, third, etc., optimal state sequences (difficult ?)
  - The Viterbi algorithm also can be illustrated in a trellis framework similar to the one for the forward algorithm
    - State-time trellis diagram

1. R. Bellman, "On the Theory of Dynamic Programming," *Proceedings of the National Academy of Sciences*, 1952
2. A.J. Viterbi , "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory* ,13 (2), 1967.

Basic Problem 2 of HMM

- The Viterbi Algorithm (cont.)

- Algorithm

  Find a best state sequence $\boldsymbol{S}=(s_1,s_2,..,s_T)$ for a given observation $\boldsymbol{O}=(\boldsymbol{o}_1,\boldsymbol{o}_2,..,\boldsymbol{o}_T)$?

  Define a new variable

$$\delta_t(i)=\max_{s_1,s_2,..,s_{t-1}} P\left[s_1,s_2,..,s_{t-1},s_t=i,\boldsymbol{o}_1,\boldsymbol{o}_2,..,\boldsymbol{o}_t \middle| \lambda\right]$$

  = the best score along a single path at time t, which accounts for the first t observation and ends in state i

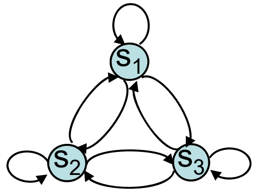
  By induction $\therefore \delta_{t+1}(j)=\left[\max_{1\le i\le N}\delta_t(i)a_{ij}\right]b_j(\boldsymbol{o}_{t+1})$

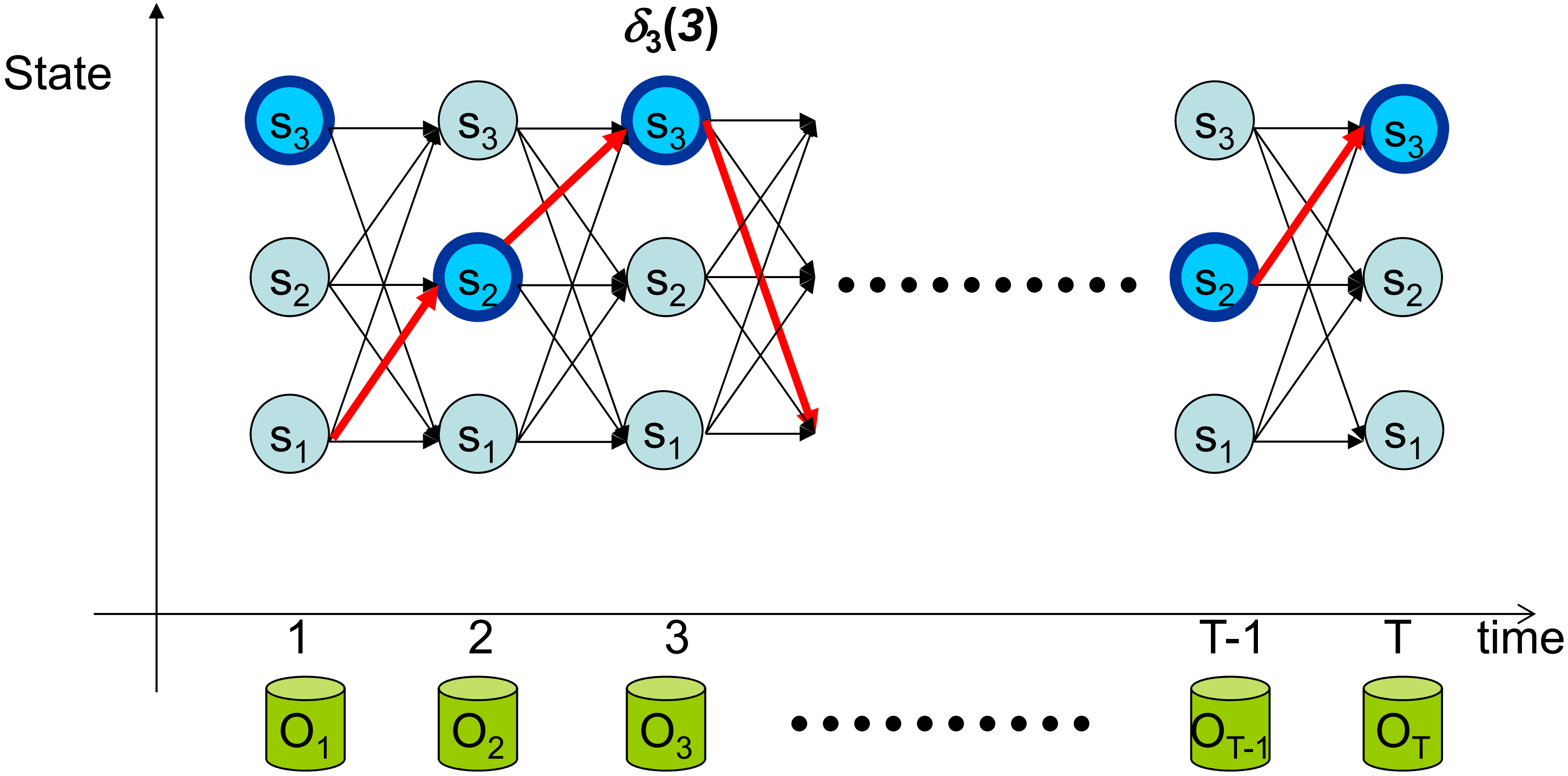
$$\psi_{t+1}(j)=\arg\max_{1\le i\le N}\delta_t(i)a_{ij} \quad \text{.... For backtracing}$$

  We can backtrace from $s_T^*=\arg\max_{1\le i\le N}\delta_T(i)$

  – Complexity: $O(N^2T)$

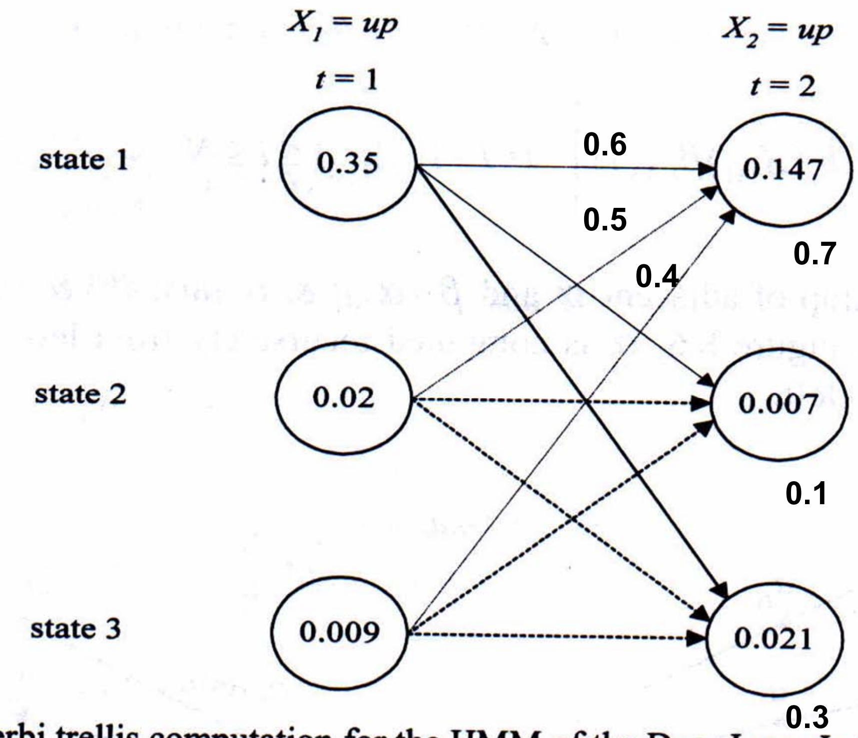
Basic Problem 2 of HMM
- The Viterbi Algorithm (cont.)





Basic Problem 2 of HMM
- The Viterbi Algorithm (cont.)

- A three-state Hidden Markov Model for the *Dow Jones Industrial average*



(0.6*0.35)*0.7
=0.147

Figure 8.5 The Viterbi trellis computation for the HMM of the Dow Jones Industrial average.

Basic Problem 2 of HMM

- The Viterbi Algorithm (cont.)

- Algorithm in the logarithmic form

Find a best state sequence $\boldsymbol{S}=(s_1,s_2,..,s_T)$ for a given observation $\boldsymbol{O}=(\boldsymbol{o}_1,\boldsymbol{o}_2,..,\boldsymbol{o}_T)$?

Define a new variable

$$\delta_t(i)=\max_{s_1,s_2,..,s_{t-1}} \log P[s_1,s_2,..,s_{t-1},s_t=i,\boldsymbol{o}_1,\boldsymbol{o}_2,..,\boldsymbol{o}_t|\lambda]$$

= the best score along a single path at time t, which accounts for the first t observation and ends in state i

By induction $\therefore \delta_{t+1}(j)=\left[\max_{1\le i\le N}(\delta_t(i)+\log a_{ij})\right]+\log b_j(\boldsymbol{o}_{t+1})$

$$\psi_{t+1}(j)=\arg\max_{1\le i\le N}(\delta_t(i)+\log a_{ij}) \text{ .... For backtracing}$$

We can backtrace from $s_T^*=\arg\max_{1\le i\le N}\delta_T(i)$

Homework 1

- A three-state Hidden Markov Model for the *Dow Jones Industrial average*

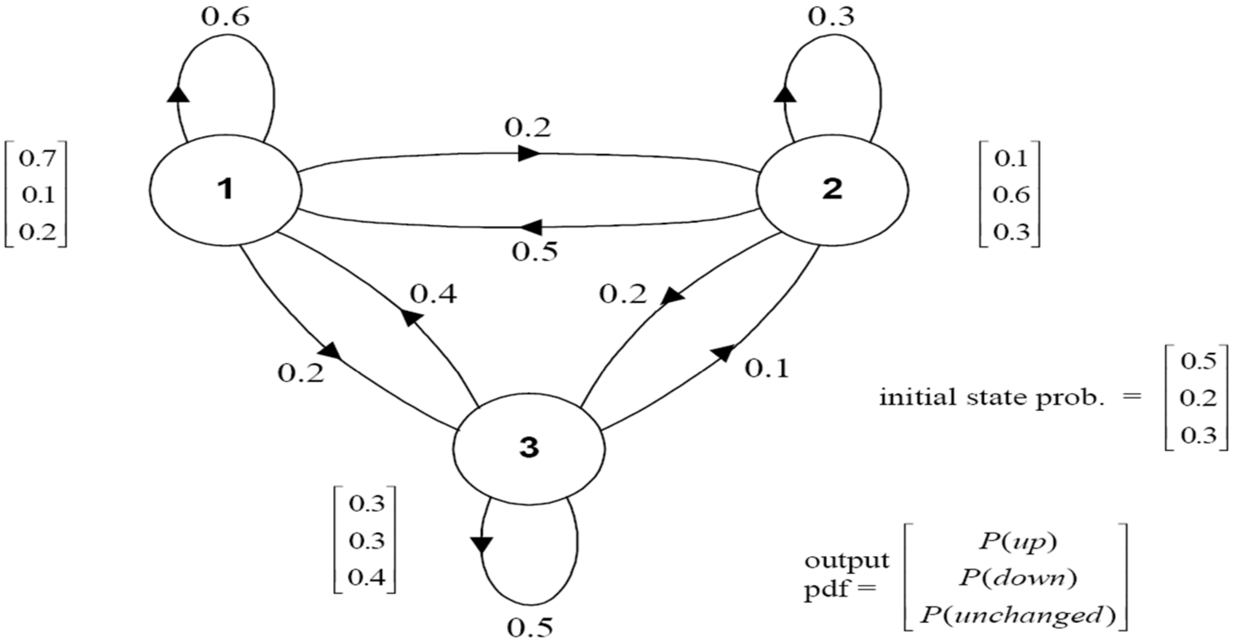
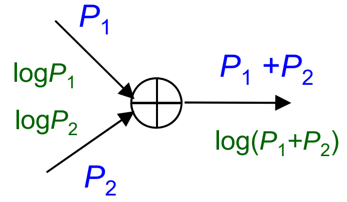


Figure 8.2 A hidden Markov model for the Dow Jones Industrial average. The three states no longer have deterministic meanings as the Markov chain illustrated in Figure 8.1.

  - *Find the probability:*
    $P(up, up, unchanged, down, unchanged, down, up|\lambda)$
  - *Fnd the optimal state sequence of the model which generates the observation sequence:* (*up, up, unchanged, down, unchanged, down, up*)

Probability Addition in F-B Algorithm

- In Forward-backward algorithm, operations usually implemented in logarithmic domain



- Assume that we want to add P_1 and P_2

$$\text{if } P_1 \geq P_2$$

$$\log_b(P_1 + P_2) = \log P_1 + \log_b\left(1 + b^{\log_b P_2 - \log_b P_1}\right)$$

$$\text{else}$$

$$\log_b(P_1 + P_2) = \log P_2 + \log_b\left(1 + b^{\log_b P_1 - \log_b P_2}\right)$$

The values of $\log_b(1 + b^x)$ can be saved in in a table to speedup the operations

Probability Addition in F-B Algorithm (cont.)

- An example code

```
#define LZERO  (-1.0E10)   // ~log(0)
#define LSMALL (-0.5E10)   // log values < LSMALL are set to LZERO
#define minLogExp  -log(-LZERO) // ~=-23
double LogAdd(double x, double y)
{
 double temp,diff,z;
   if (x<y)
   {
      temp = x; x = y; y = temp;
   }
   diff = y-x; //notice that diff <= 0
   if (diff<minLogExp) // if y’ is far smaller than x’
      return  (x<LSMALL) ?  LZERO:x;
   else
    {
      z = exp(diff);
      return x+log(1.0+z);
   }
}
```

Basic Problem 3 of HMM

Intuitive View

- How to adjust (re-estimate) the model parameter λ=(***A***,***B***,π) to maximize $P(\mathbf{O}_1,\ldots,\mathbf{O}_L|\lambda)$ or log$P(\mathbf{O}_1,\ldots,\mathbf{O}_L|\lambda)$?
  - Belonging to a typical problem of "inferential statistics"
  - The most difficult of the three problems, because there is no known analytical method that maximizes the joint probability of the training data in a close form

$$\log P(\mathbf{O}_1,\mathbf{O}_2,\ldots,\mathbf{O}_L|\lambda)=\log\prod_{l=1}^{L}P(\mathbf{O}_l|\lambda)$$

$$=\sum_{l=1}^{L}\log P(\mathbf{O}_l|\lambda)=\sum_{l=1}^{R}\boxed{\log\sum_{all\,\mathbf{S}}P(\mathbf{S}|\lambda)P(\mathbf{O}_l|\mathbf{S},\lambda)}$$

The "log of sum" form is difficult to deal with

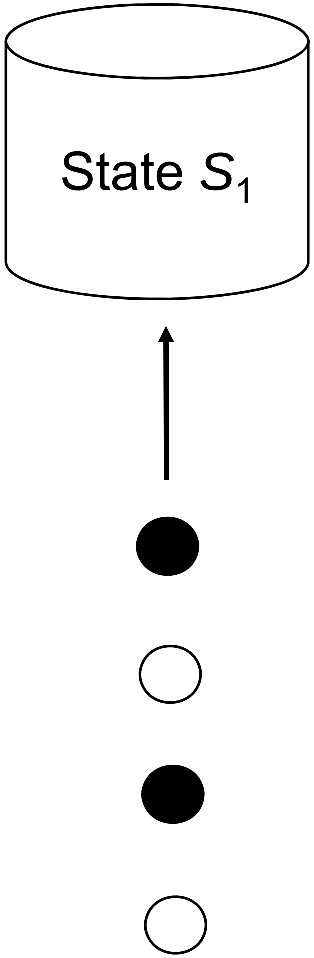
- Suppose that we have L training utterances for the HMM

- $\mathbf{S}$: a possible state sequence of the HMM

  - The data is incomplete because of the hidden state sequences
  - Well-solved by the *Baum-Welch* (known as *forward-backward*) algorithm and *EM* (*Expectation-Maximization*) algorithm
    - Iterative update and improvement
    - Based on Maximum Likelihood (ML) criterion

Maximum Likelihood (ML) Estimation: A Schematic Depiction (1/2)

- Hard Assignment
  - Given the data follow a multinomial distribution



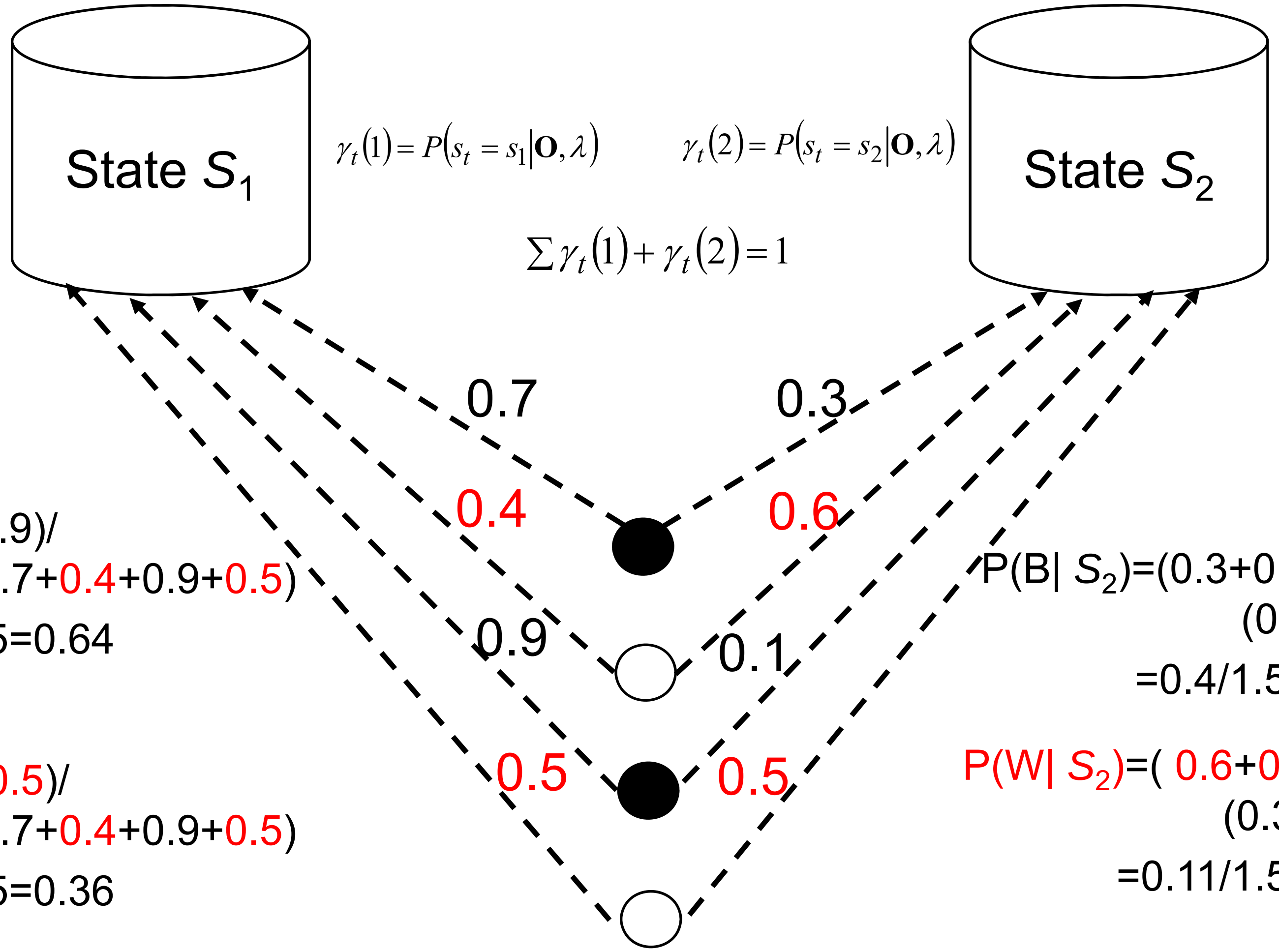
State S_1

$P(B| S_1)=2/4=0.5$

$P(W| S_1)=2/4=0.5$

Maximum Likelihood (ML) Estimation: A Schematic Depiction (1/2)

- Soft Assignment
  - Given the data follow a multinomial distribution
  - Maximize the likelihood of the data given the alignment



$\gamma_t(1) = P(s_t = s_1 | \mathbf{O}, \lambda)$

$\gamma_t(2) = P(s_t = s_2 | \mathbf{O}, \lambda)$

$\sum \gamma_t(1) + \gamma_t(2) = 1$

P(B| S_1)=(0.7+0.9)/
(0.7+0.4+0.9+0.5)
=1.6/2.5=0.64

P(W| S_1)=(0.4+0.5)/
(0.7+0.4+0.9+0.5)
=0.9/2.5=0.36

P(B| S_2)=(0.3+0.1)/
(0.3+0.6+0.1+0.5)
=0.4/1.5=0.27

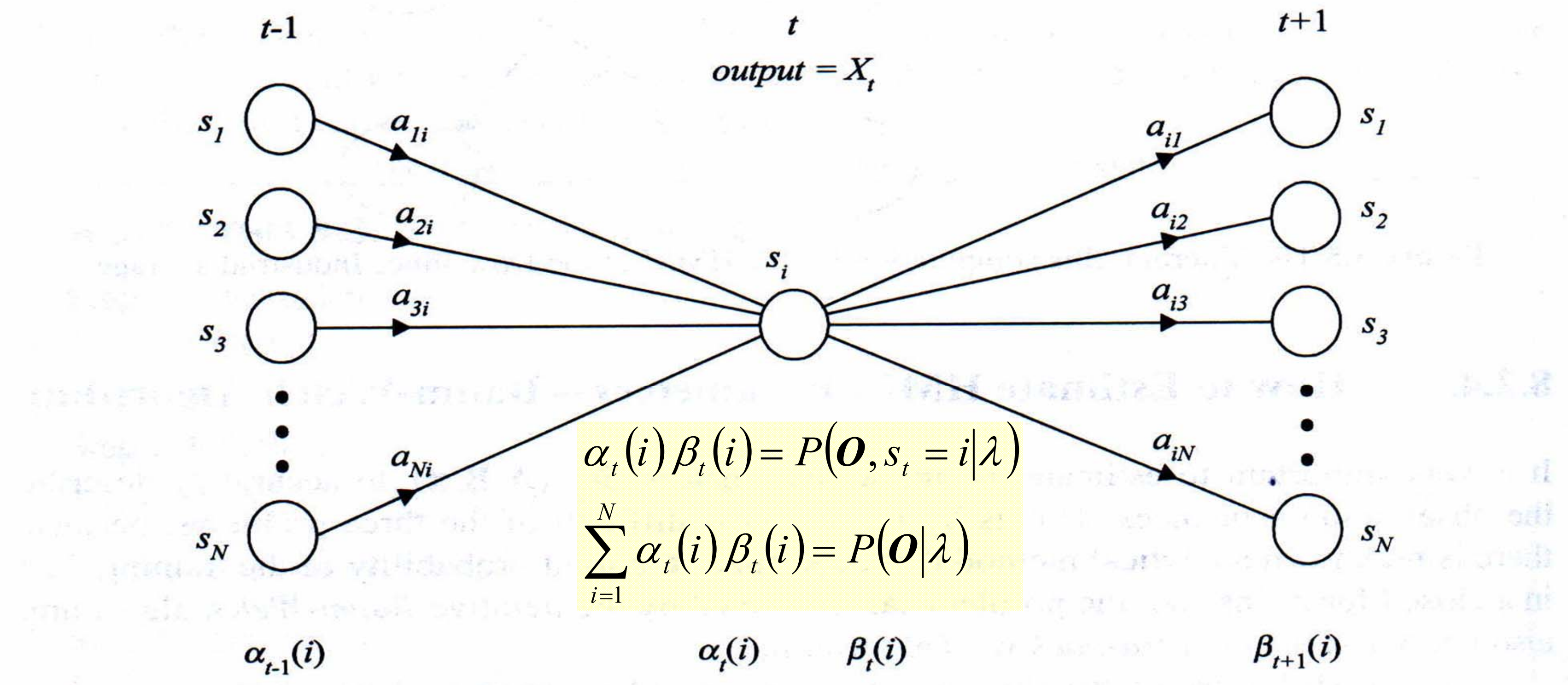
P(W| S_2)=(0.6+0.5)/
(0.3+0.6+0.1+0.5)
=0.11/1.5=0.73

Basic Problem 3 of HMM
Intuitive View (cont.)

- Relationship between the forward and backward variables

$$\alpha_t(i) = P(\boldsymbol{o}_1, \boldsymbol{o}_2, ..., \boldsymbol{o}_t, s_t = i | \lambda) = \left[\sum_{j=1}^{N} \alpha_{t-1}(j) a_{ji} \right] b_i(\boldsymbol{o}_t)$$

$$\beta_t(i) = P(\boldsymbol{o}_{t+1}, \boldsymbol{o}_{t+2}, ..., \boldsymbol{o}_T | s_t = i, \lambda) = \sum_{j=1}^{N} \beta_{t+1}(j) b_j(\boldsymbol{o}_{t+1}) a_{ij}$$

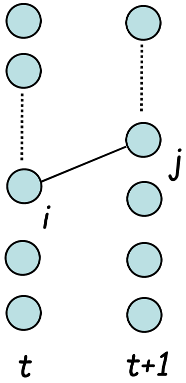


$$\alpha_t(i)\,\beta_t(i) = P(\boldsymbol{O}, s_t = i | \lambda)$$

$$\sum_{i=1}^{N} \alpha_t(i)\,\beta_t(i) = P(\boldsymbol{O} | \lambda)$$

Figure 8.6 The relationship of α_{t-1} and α_t and β_t and β_{t+1} in the forward-backward algorithm.

Basic Problem 3 of HMM
Intuitive View (cont.)



- Define a new variable:

$$\xi_t(i,j) = P(s_t = i, s_{t+1} = j | \boldsymbol{O}, \boldsymbol{\lambda})$$

  - Probability being at state *i* at time *t* and at state *j* at time *t+1*

$$p(A|B) = \frac{p(A,B)}{P(B)}$$

$$\xi_t(i,j) = \frac{P(s_t = i, s_{t+1} = j, \boldsymbol{O} | \boldsymbol{\lambda})}{P(\boldsymbol{O}|\boldsymbol{\lambda})}$$

$$= \frac{\alpha_t(i) a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j)}{P(\boldsymbol{O}|\boldsymbol{\lambda})} = \frac{\alpha_t(i) a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j)}{\sum_{m=1}^{N}\sum_{n=1}^{N} \alpha_t(m) a_{mn} b_n(\boldsymbol{o}_{t+1}) \beta_{t+1}(n)}$$

- Recall the posteriori probability variable:

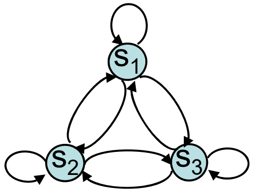
$$\gamma_t(i) = P(s_t = i | \boldsymbol{O}, \lambda)$$

Note : $\gamma_t(i)$ also can be represente d as $\frac{\alpha_t(i)\beta_t(i)}{\sum_{m=1}^{N} \alpha_t(m)\beta_t(m)}$

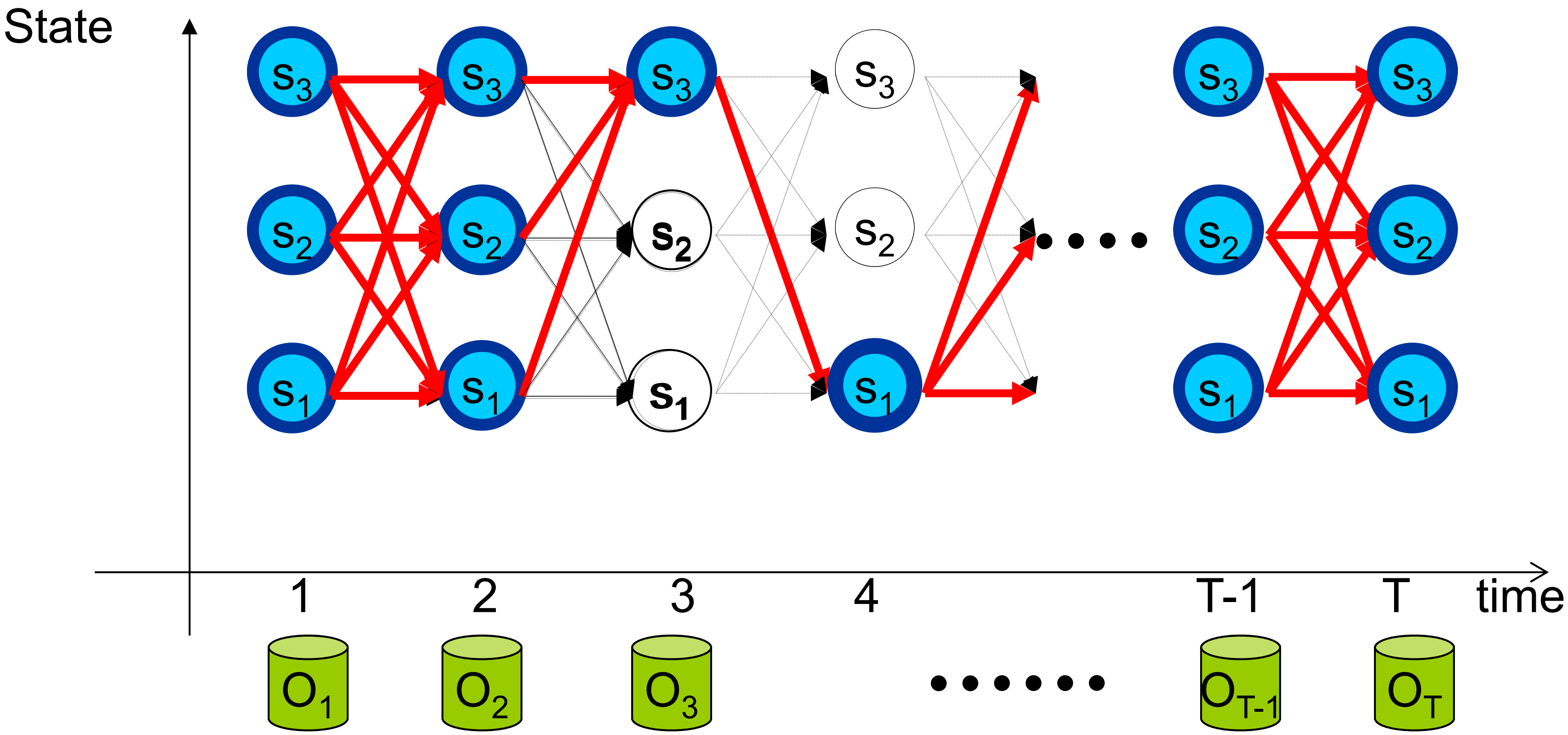
$$\gamma_t(i) = \sum_{j=1}^{N} P(s_t = i, s_{t+1} = j | \boldsymbol{O}, \lambda) = \sum_{j=1}^{N} \xi_t(i,j) \quad (\text{for } t < T)$$

Basic Problem 3 of HMM

Intuitive View (cont.)



- $P(s_3 = 3, s_4 = 1, \mathbf{O} \mid \lambda) = \alpha_3(3) * a_{31} * b_1(o_4) * \beta_1(4)$



Basic Problem 3 of HMM

Intuitive View (cont.)

- $\xi_t(i,j) = P(s_t = i, s_{t+1} = j | \boldsymbol{O}, \boldsymbol{\lambda})$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{expected number of transitions from state } i \text{ to state } j \text{ in } \boldsymbol{O}$$

- $\gamma_t(i) = P(s_t = i | \boldsymbol{O}, \lambda)$

$$\sum_{t=1}^{T-1} \gamma_t(i) = \sum_{t=1}^{T-1} \sum_{j=1}^{N} \xi_t(i,j) = \text{expected number of transitions from state } i \text{ in } \boldsymbol{O}$$

- A set of reasonable re-estimation formula for {***A***,π} is

$$\begin{aligned} \overline{\pi}_i &= \text{expected freqency (number of times) in state } i \text{ at time } t = 1 \\ &= \gamma_1(i) \end{aligned}$$

$$\overline{a}_{ij} = \frac{\text{expected number of transitio n from state } i \text{ to state } j}{\text{expected number of transitio n from state } i} = \frac{\sum_{t=1}^{T\text{-}1} \xi_t(i,j)}{\sum_{t=1}^{T\text{-}1} \gamma_t(i)}$$

Formulae for Single Training Utterance

Basic Problem 3 of HMM

Intuitive View (cont.)

- A set of reasonable re-estimation formula for {*B*} is
  - For discrete and finite observation $b_j(\mathbf{v}_k)=P(\mathbf{o}_t=\mathbf{v}_k|\mathbf{s}_t=j)$

$$\bar{b}_j(\boldsymbol{v}_k)=\bar{P}(\boldsymbol{o}=\boldsymbol{v}_k|s=j)=\frac{\text{expected number of times in state } j \text{ and observing symbol } \boldsymbol{v}_k}{\text{expected number of times in state } j}=\frac{\sum_{\substack{t=1\\ \text{such that } \boldsymbol{o}=\boldsymbol{v}_k}}^{T}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)}$$

  - For continuous and infinite observation $b_j(\mathbf{v})=f_{\mathbf{O}|\mathbf{S}}(\mathbf{o}_t=\mathbf{v}|\mathbf{s}_t=j)$,

$$\bar{b}_j(\boldsymbol{v})=\sum_{k=1}^{M}\bar{c}_{jk}N(\boldsymbol{v};\bar{\boldsymbol{\mu}}_{jk},\bar{\boldsymbol{\Sigma}}_{jk})=\sum_{k=1}^{M}\bar{c}_{jk}\left(\frac{1}{\left(\sqrt{2\pi}\right)^{L}\left|\bar{\boldsymbol{\Sigma}}_{jk}\right|^{1/2}}\exp\left(-\frac{1}{2}(\boldsymbol{v}-\bar{\boldsymbol{\mu}}_{jk})^{t}\bar{\Sigma}_{jk}^{-1}(\boldsymbol{v}-\bar{\boldsymbol{\mu}}_{jk})\right)\right)$$

Modeled as a mixture of multivariate Gaussian distributions

Basic Problem 3 of HMM
Intuitive View (cont.)

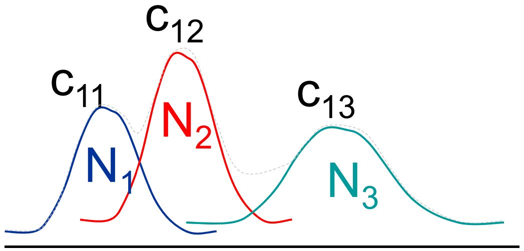
$$p(A|B)=\frac{p(A,B)}{P(B)}$$

- For continuous and infinite observation (Cont.)
  - Define a new variable $\gamma_t(j,k)$
    - $\gamma_t(j,k)$ is the probability of being in state *j* at time *t* with the *k*-th mixture component accounting for ***o***$_t$

$$\begin{aligned}
\gamma_t(j,k) &= P(s_t=j, m_t=k|\mathbf{O},\boldsymbol{\lambda})\\
&= P(s_t=j|\mathbf{O},\boldsymbol{\lambda})P(m_t=k|s_t=j,\mathbf{O},\boldsymbol{\lambda})\\
&= \gamma_t(j)P(m_t=k|s_t=j,\mathbf{O},\boldsymbol{\lambda})\\
&= \gamma_t(j)\frac{p(m_t=k,\mathbf{O}|s_t=j,\boldsymbol{\lambda})}{p(\mathbf{O}|s_t=j,\boldsymbol{\lambda})}\\
&= \gamma_t(j)\frac{P(m_t=k|s_t=j,\boldsymbol{\lambda})p(\mathbf{O}|s_t=j,m_t=k,\boldsymbol{\lambda})}{p(\mathbf{O}|s_t=j,\boldsymbol{\lambda})}
\end{aligned}$$

...... (observation - independent assumption is applied)

$$\begin{aligned}
&= \gamma_t(j)\frac{P(m_t=k|s_t=j,\boldsymbol{\lambda})p(\mathbf{o}_t|s_t=j,m_t=k,\boldsymbol{\lambda})}{p(\mathbf{o}_t|s_t=j,\boldsymbol{\lambda})}\\
&= \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{s=1}^{N}\alpha_t(s)\beta_t(s)}\right]\left[\frac{c_{jk}N(\mathbf{o}_t;\boldsymbol{\mu}_{jk},\boldsymbol{\Sigma}_{jk})}{\sum_{m=1}^{M}c_{jm}N(\mathbf{o}_t;\boldsymbol{\mu}_{jm},\boldsymbol{\Sigma}_{jm})}\right]
\end{aligned}$$



Distribution for State 1

Note: $\gamma_t(j)=\sum_{m=1}^{M}\gamma_t(j,m)$

Basic Problem 3 of HMM
Intuitive View (cont.)

- For continuous and infinite observation (Cont.)

$$\bar{c}_{jk} = \frac{\text{expected number of times in state } j \text{ and mixture } k}{\text{expected number of times in state } j} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T}\sum_{m=1}^{M} \gamma_t(j,m)}$$

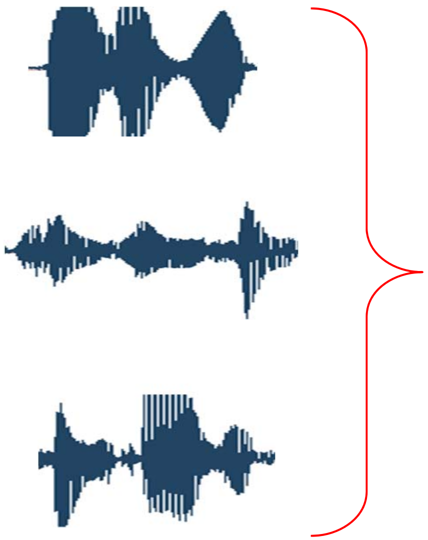
$$\bar{\boldsymbol{\mu}}_{jk} = \text{weighted average (mean) of observatio ns at state } j \text{ and mixture } k = \frac{\sum_{t=1}^{T} \gamma_t(j,k)\cdot \boldsymbol{o}_t}{\sum_{t=1}^{T} \gamma_t(j,k)}$$

$$\bar{\boldsymbol{\Sigma}}_{jk} = \text{weighted covariance of observations at state } j \text{ and mixture } k$$
$$= \frac{\sum_{t=1}^{T} \gamma_t(j,k)\cdot(\boldsymbol{o}_t - \bar{\boldsymbol{\mu}}_{jk})(\boldsymbol{o}_t - \bar{\boldsymbol{\mu}}_{jk})^t}{\sum_{t=1}^{T} \gamma_t(j,k)}$$

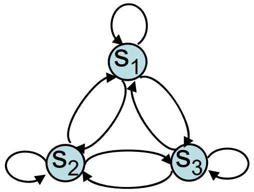
Formulae for Single Training Utterance

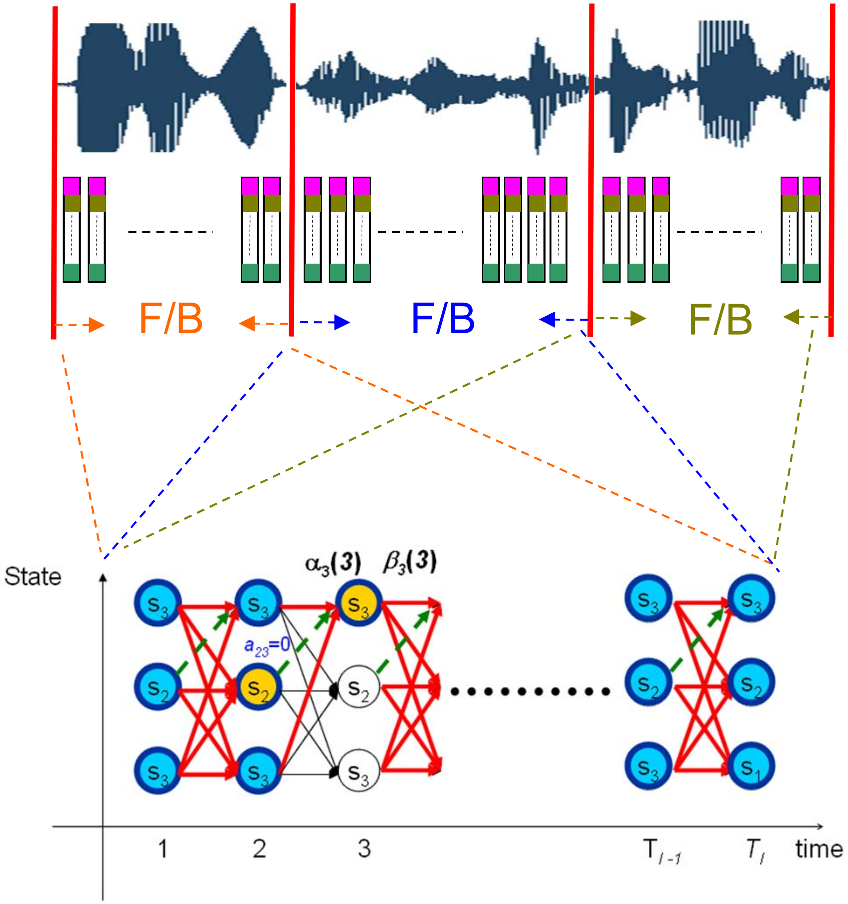
Basic Problem 3 of HMM
Intuitive View (cont.)

- Multiple Training Utterances



台師大





Basic Problem 3 of HMM

Intuitive View (cont.)

– For continuous and infinite observation (Cont.)

$$\overline{\pi}_i = \text{expected freqency (number of times) in state } i \text{ at time } (t=1) = \frac{1}{L}\sum_{l=1}^{L}\gamma_1^l(i)$$

$$\overline{a}_{ij} = \frac{\text{expected number of transition from state } i \text{ to state } j}{\text{expected number of transition from state } i} = \frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l\text{-}1}\xi_t^l(i,j)}{\sum_{l=1}^{L}\sum_{t=1}^{T_l\text{-}1}\gamma_t^l(i)}$$

$$\overline{c}_{jk} = \frac{\text{expected number of times in state } j \text{ and mixture } k}{\text{expected number of times in state } j} = \frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l}\gamma_t^l(j,k)}{\sum_{l=1}^{L}\sum_{t=1}^{T_l}\sum_{m=1}^{M}\gamma_t^l(j,m)}$$

$$\overline{\boldsymbol{\mu}}_{jk} = \text{weighted average (mean) of observations at state } j \text{ and mixture } k = \frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l}\gamma_t^l(j,k)\cdot\mathbf{o}_t}{\sum_{l=1}^{L}\sum_{t=1}^{T_l}\gamma_t^l(j,k)}$$

$$\overline{\boldsymbol{\Sigma}}_{jk} = \text{weighted covariance of observations at state } j \text{ and mixture } k$$

$$= \frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l}\gamma_t^l(j,k)\cdot\left(\mathbf{o}_t-\overline{\boldsymbol{\mu}}_{jk}\right)\left(\mathbf{o}_t-\overline{\boldsymbol{\mu}}_{jk}\right)^t}{\sum_{l=1}^{L}\sum_{t=1}^{T_l}\gamma_t^l(j,k)}$$

Formulae for Multiple (L) Training Utterances

Basic Problem 3 of HMM

Intuitive View (cont.)

- For discrete and finite observation (cont.)

$$\bar{\pi}_i = \text{expected freqency (number of times) in state } i \text{ at time } (t=1) = \frac{1}{L}\sum_{l=1}^{L}\gamma_1^l(i)$$

$$\bar{a}_{ij} = \frac{\text{expected number of transition from state } i \text{ to state } j}{\text{expected number of transition from state } i} = \frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l-1}\xi_t^l(i,j)}{\sum_{l=1}^{L}\sum_{t=1}^{T_l-1}\gamma_t^l(i)}$$

$$\bar{b}_j(\mathbf{v}_k) = \bar{P}(\mathbf{o}=\mathbf{v}_k|s=j) = \frac{\text{expected number of times in state } j \text{ and observing symbol } \mathbf{v}_k}{\text{expected number of times in state } j} = \frac{\sum_{l=1}^{L}\sum_{\substack{\text{t}=1 \\ \text{such that } \mathbf{o}=\mathbf{v}_k}}^{T_l}\gamma_t^l(j)}{\sum_{l=1}^{L}\sum_{\text{t}=1}^{T_l}\gamma_t^l(j)}$$

Formulae for Multiple (*L*) Training Utterances

Semicontinuous HMMs

- The HMM state mixture density functions are tied together across all the models to form a set of shared kernels
  - The semicontinuous or tied-mixture HMM

$$b_j(\boldsymbol{o}) = \sum_{k=1}^{M} b_j(k) f(\boldsymbol{o}|v_k) = \sum_{k=1}^{M} b_j(k) N(\boldsymbol{o}, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

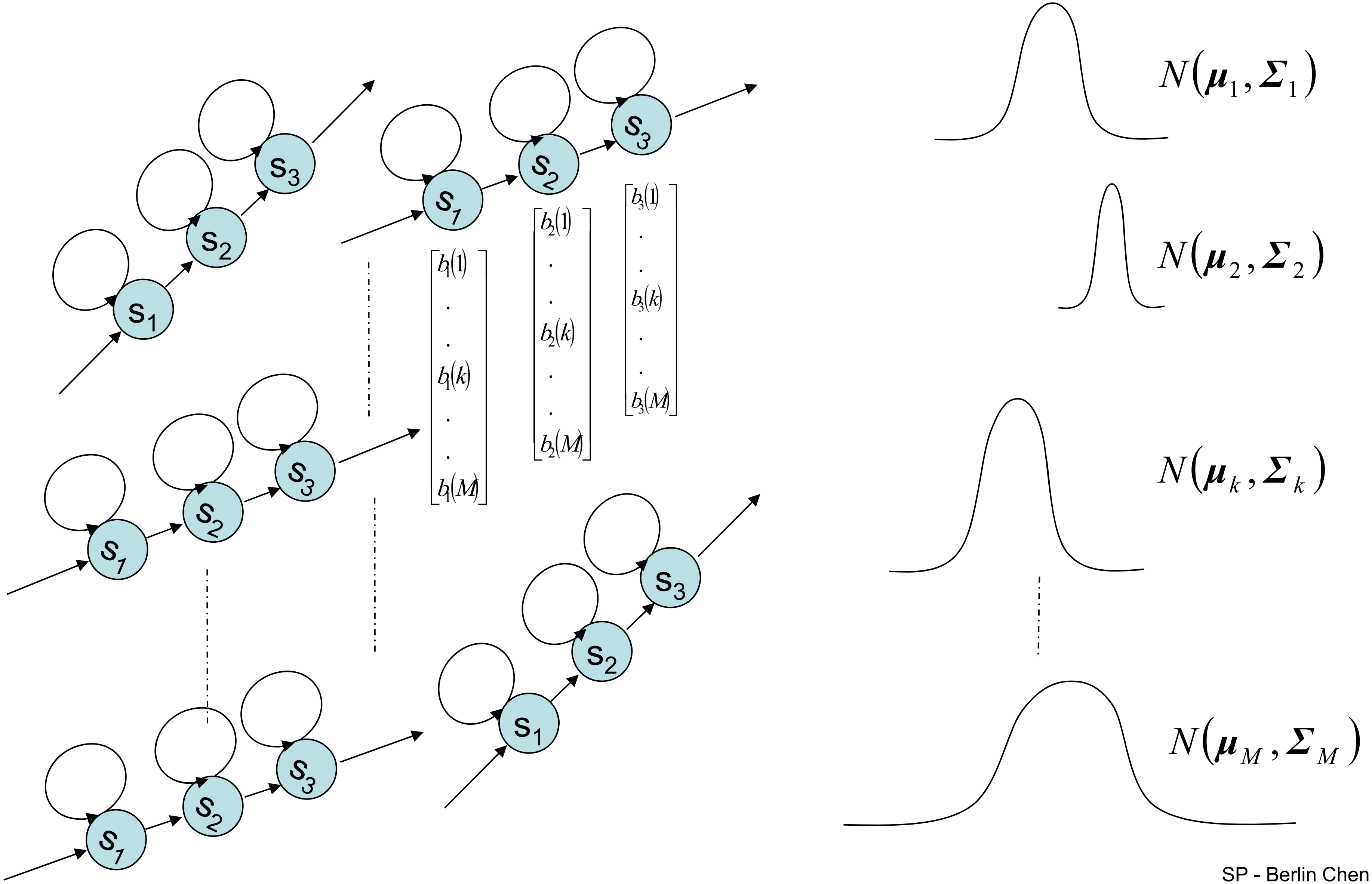
state output
Probability of state *j*

k-th mixture weight
t of state *j*
(discrete, model-dependent)

k-th mixture density function or *k*-th codeword
(shared across HMMs, *M* is very large)

  - A combination of the discrete HMM and the continuous HMM
    - A combination of *discrete* model-dependent weight coefficients and *continuous* model-independent codebook probability density functions
  - Because *M* is large, we can simply use the *L* most significant values $f(\boldsymbol{o}|v_k)$
    - Experience showed that *L* is *1~3%* of *M* is adequate
  - Partial tying of $f(\boldsymbol{o}|v_k)$ for different phonetic class

Semicontinuous HMMs (cont.)



HMM Topology

- Speech is time-evolving non-stationary signal
  - Each HMM state has the ability to capture some quasi-stationary segment in the non-stationary speech signal
  - A *left-to-right* topology is a natural candidate to model the speech signal (also called the “beads-on-a-string” model)

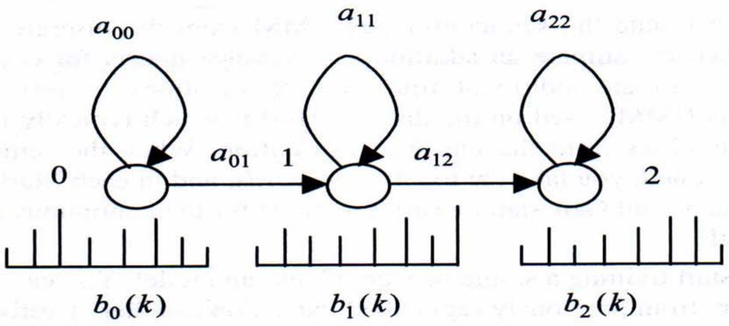
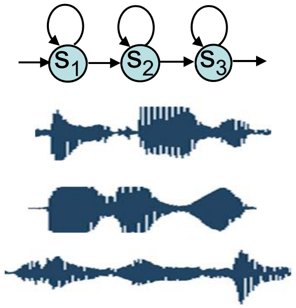


Figure 8.8 A typical hidden Markov model used to model phonemes. There are three states (0-2) and each state has an associated output probability distribution.

  - It is general to represent a phone using 3~5 states (English) and a syllable using 6~8 states (Mandarin Chinese)



Initialization of HMM

- A good initialization of HMM training :
  Segmental K-Means Segmentation into States
  - Assume that we have a training set of observations and an initial estimate of all model parameters
  - Step 1 : The set of training observation sequences is segmented into states, based on the initial model (finding the optimal state sequence by *Viterbi* Algorithm)
  - Step 2 :
    - For discrete density HMM (using M-codeword codebook)

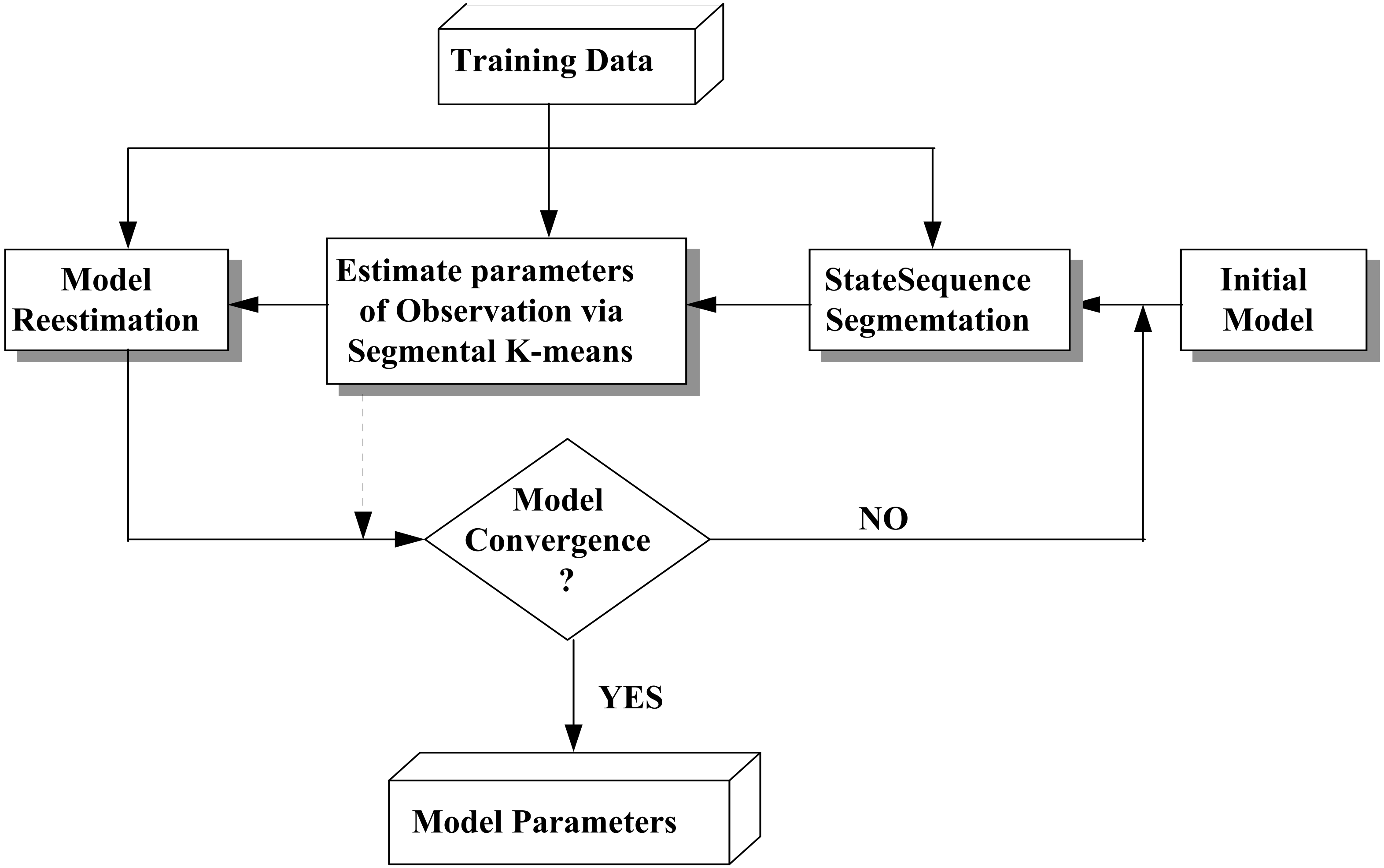
$$\bar{b}_j(k) = \frac{\text{the number of vectors with codebook index } k \text{ in state } j}{\text{the number of vectors in state } j}$$

    - For continuous density HMM (M Gaussian mixtures per state)

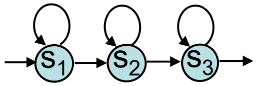
$$\begin{aligned}
&\Rightarrow \text{cluster the observation vectors within each state } j \text{ into a set of } M \text{ clusters} \\
&\bar{w}_{jm} = \text{number of vectors classified in cluster } m \text{ of state } j \\
&\qquad\quad \text{divided by the number of vectors in state } j \\
&\bar{\mu}_{jm} = \text{sample mean of the vectors classified in cluster } m \text{ of state } j \\
&\bar{\Sigma}_{jm} = \text{sample covariance matrix of the vectors classified in cluster } m \text{ of state } j
\end{aligned}$$

  - Step 3: Evaluate the model score
    If the difference between the previous and current model scores is greater than a threshold, go back to Step 1, otherwise stop, the initial model is generated

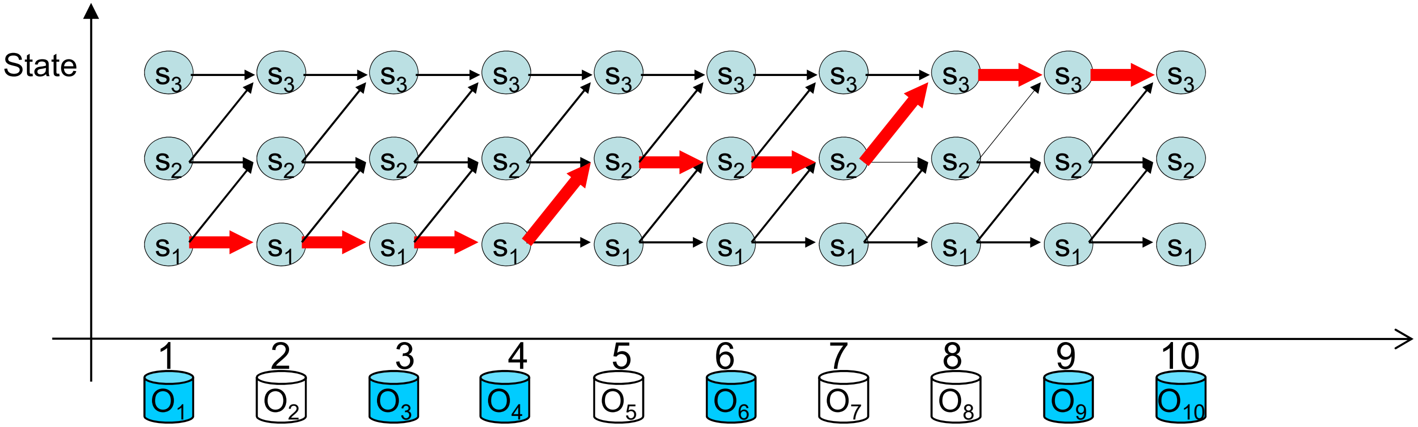
Initialization of HMM (cont.)



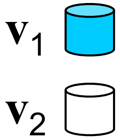
Initialization of HMM (cont.)



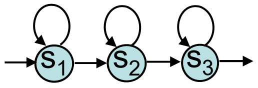
- An example for discrete HMM
  - 3 states and 2 codeword



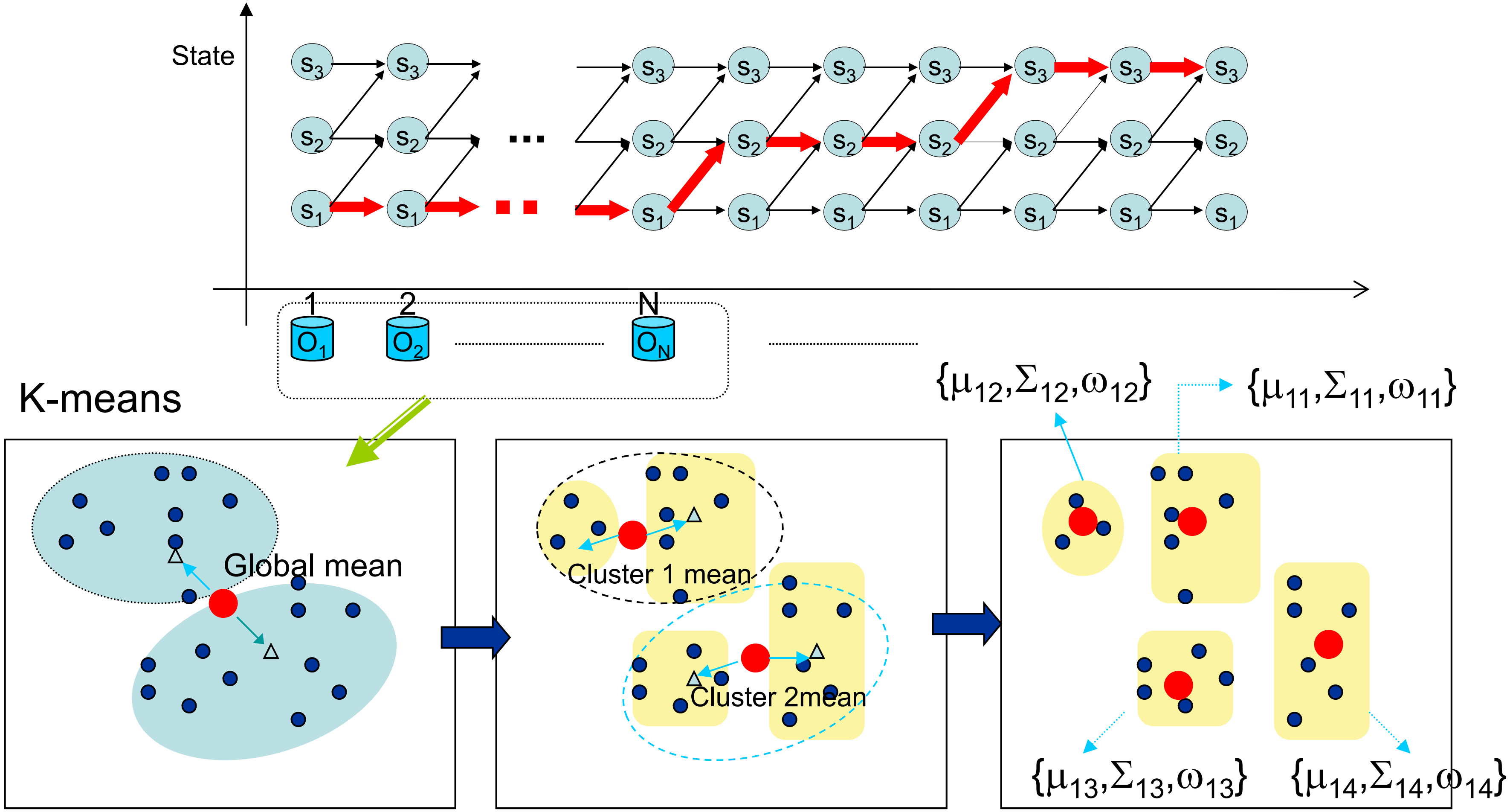
- $b_1(\mathbf{v}_1)=3/4$, $b_1(\mathbf{v}_2)=1/4$
- $b_2(\mathbf{v}_1)=1/3$, $b_2(\mathbf{v}_2)=2/3$
- $b_3(\mathbf{v}_1)=2/3$, $b_3(\mathbf{v}_2)=1/3$



Initialization of HMM (cont.)



- An example for Continuous HMM
  - 3 states and 4 Gaussian mixtures per state



Known Limitations of HMMs (1/3)

- The assumptions of conventional HMMs in Speech Processing
  - The state duration follows an exponential distribution
    - Don’t provide adequate representation of the temporal structure of speech

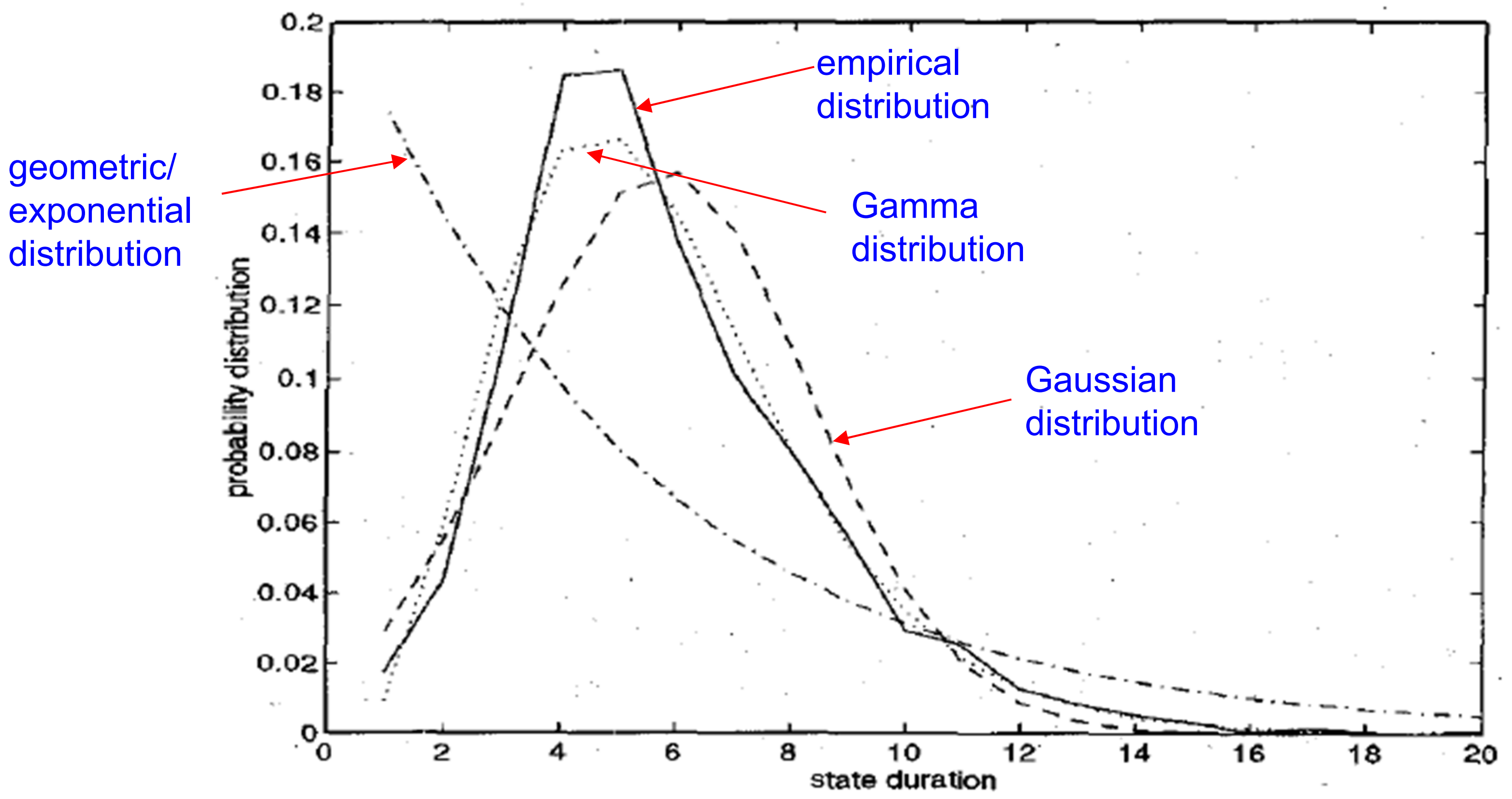
$$d_i(t) = a_{ii}^{t-1}(1 - a_{ii})$$

  - **First-order (Markov) assumption:** the state transition depends only on the origin and destination
  - **Output-independent assumption:** all observation frames are dependent on the state that generated them, not on neighboring observation frames

Researchers have proposed a number of techniques to address these limitations, albeit these solution have not significantly improved speech recognition accuracy for practical applications.

Known Limitations of HMMs (2/3)

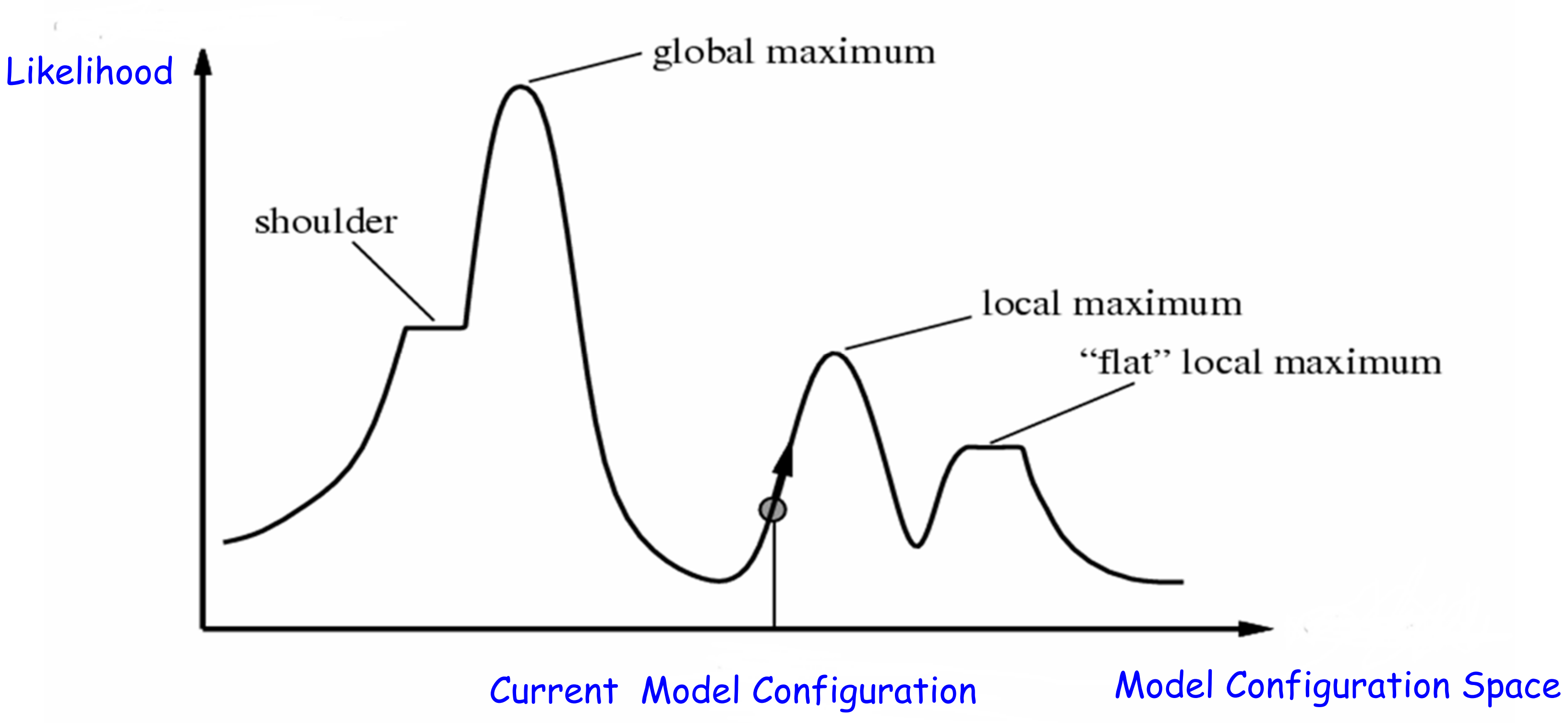
- Duration modeling



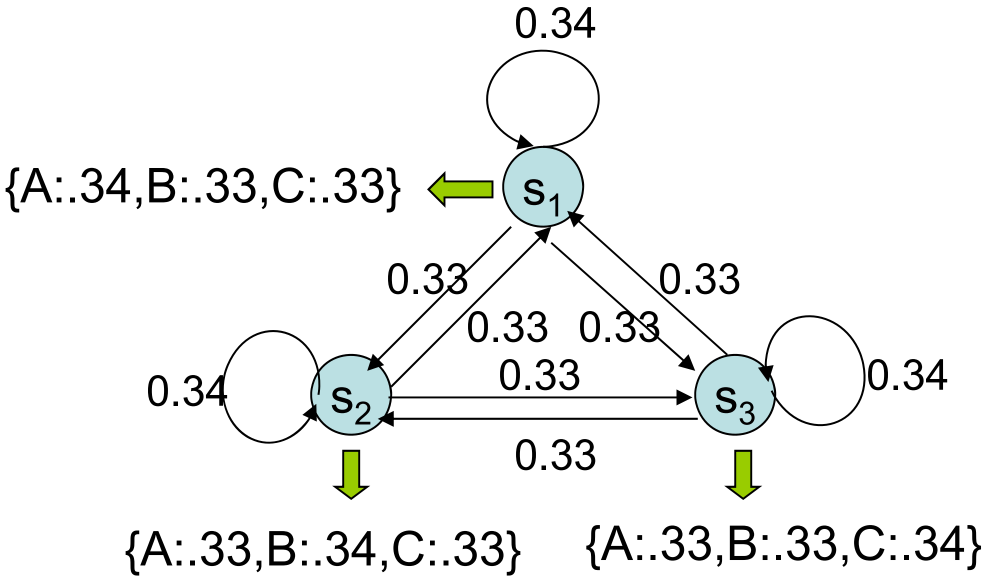
Duration distributions for the seventh state of the word "seven:" empirical distribution (solid line); Gauss fit (dashed line); gamma fit (dotted line); and (d) geometric fit (dash-dot line).

Known Limitations of HMMs (3/3)

- The HMM parameters trained by the Baum-Welch algorithm (or EM algorithm) were only locally optimized



Homework-2 (1/2)



TrainSet 1:
1. ABBCABCAABC
2. ABCABC
3. ABCA ABC
4. BBABCAB
5. BCAABCCAB
6. CACCABCA
7. CABCABCA
8. CABCA
9. CABCA

TrainSet 2:
1. BBBCCBC
2. CCBABB
3. AACCBBB
4. BBABBAC
5. CCA ABBAB
6. BBBCCBAA
7. ABBBBABA
8. CCCCC
9. BBAAA

Homework-2 (2/2)

P1. Please specify the model parameters after the first and 50th iterations of Baum-Welch training

P2. Please show the recognition results by using the above training sequences as the testing data (The so-called inside testing).

*You have to perform the recognition task with the HMMs trained from the first and 50th iterations of Baum-Welch training, respectively

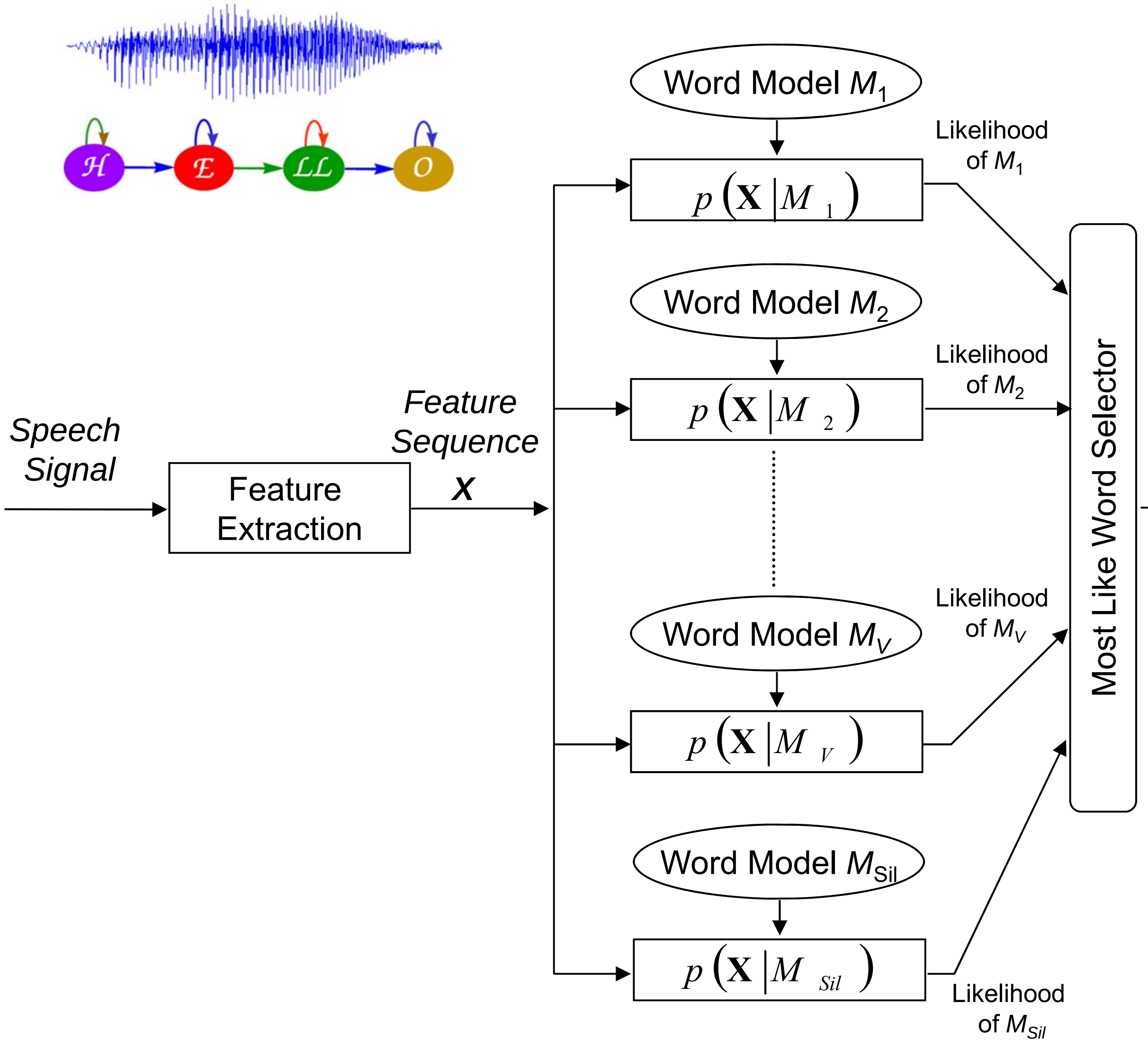
P3. Which class do the following testing sequences belong to?

ABCABCCAB

AABABCCCCBBB

P4. What are the results if Observable Markov Models were instead used in P1, P2 and P3?

Isolated Word Recognition



$$Label(\mathbf{X}) = \arg\max_k \; p(\mathbf{X}|M_k)$$

M_{ML}

Viterbi Approximation

$$Label(\mathbf{X}) = \arg\max_k \left[\max_{\mathbf{S}} \; p(\mathbf{X},\mathbf{S}|M_k)\right]$$

Measures of ASR Performance (1/2)

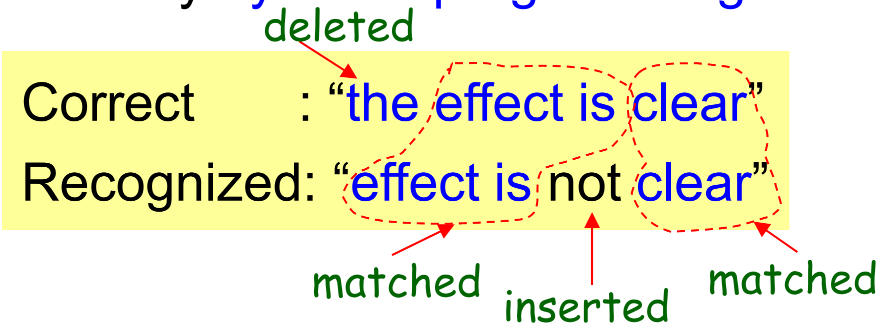
- Evaluating the performance of automatic speech recognition (ASR) systems is critical, and the Word Recognition Error Rate (WER) is one of the most important measures

- There are typically three types of word recognition errors
  - Substitution
    - An incorrect word was substituted for the correct word
  - Deletion
    - A correct word was omitted in the recognized sentence
  - Insertion
    - An extra word was added in the recognized sentence

- How to determine the minimum error rate?

Measures of ASR Performance (2/2)

- Calculate the WER by aligning the correct word string against the recognized word string
  - A maximum substring matching problem
  - Can be handled by dynamic programming
- Example:



Correct : "the effect is clear"

Recognized: "effect is not clear"

  - Error analysis: one deletion and one insertion
  - Measures: word error rate (WER), word correction rate (WCR), word accuracy rate (WAR)

Might be higher than 100%

$$\text{Word Error Rate} = 100\% \frac{\text{Sub.} + \text{Del.} + \text{Ins. words}}{\text{No. of words in the correct sentence}} = \frac{2}{4} = 50\%$$

$$\text{Word Correction Rate} = 100\% \frac{\text{Matched words}}{\text{No. of words in the correct sentence}} = \frac{3}{4} = 75\%$$

$$\text{Word Accuracy Rate} = 100\% \frac{\text{Matched - Ins. words}}{\text{No. of words in the correct sentence}} = \frac{3-1}{4} = 50\%$$

Might be negative

WER+
WAR
=100%