數位語音處理概論
Introduction to Digital Speech Processing

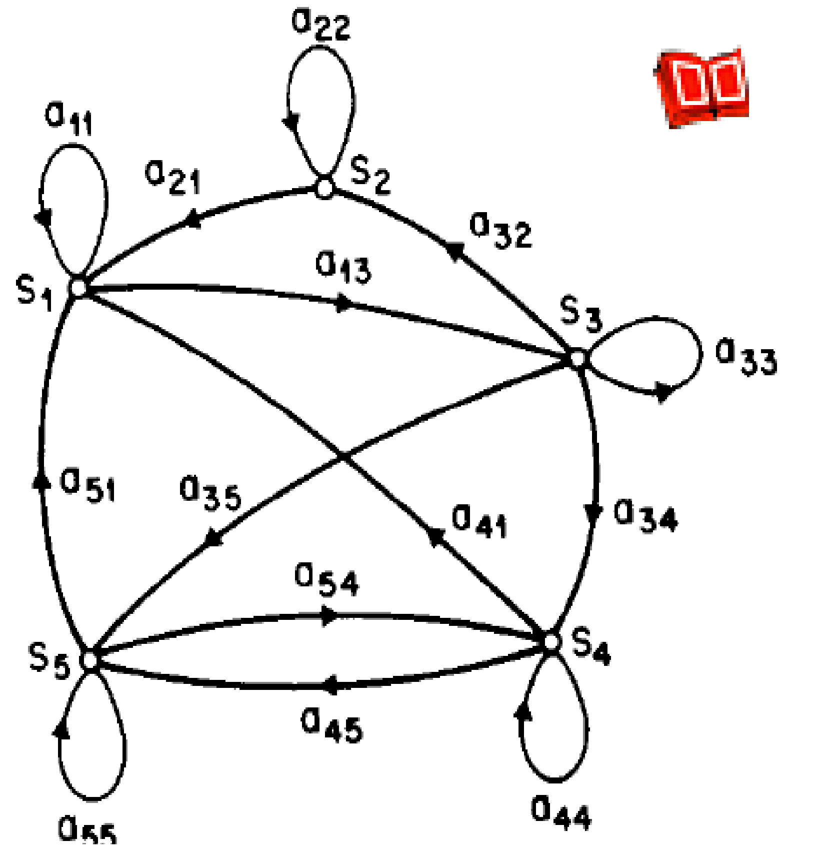
4.0 More about Hidden Markov Models

References for 4.0
1. 6.1-6.6, Rabiner and Juang, 2. 4.4.1 of Huang

授課教師：國立臺灣大學 電機工程學系 李琳山 教授

Markov Model

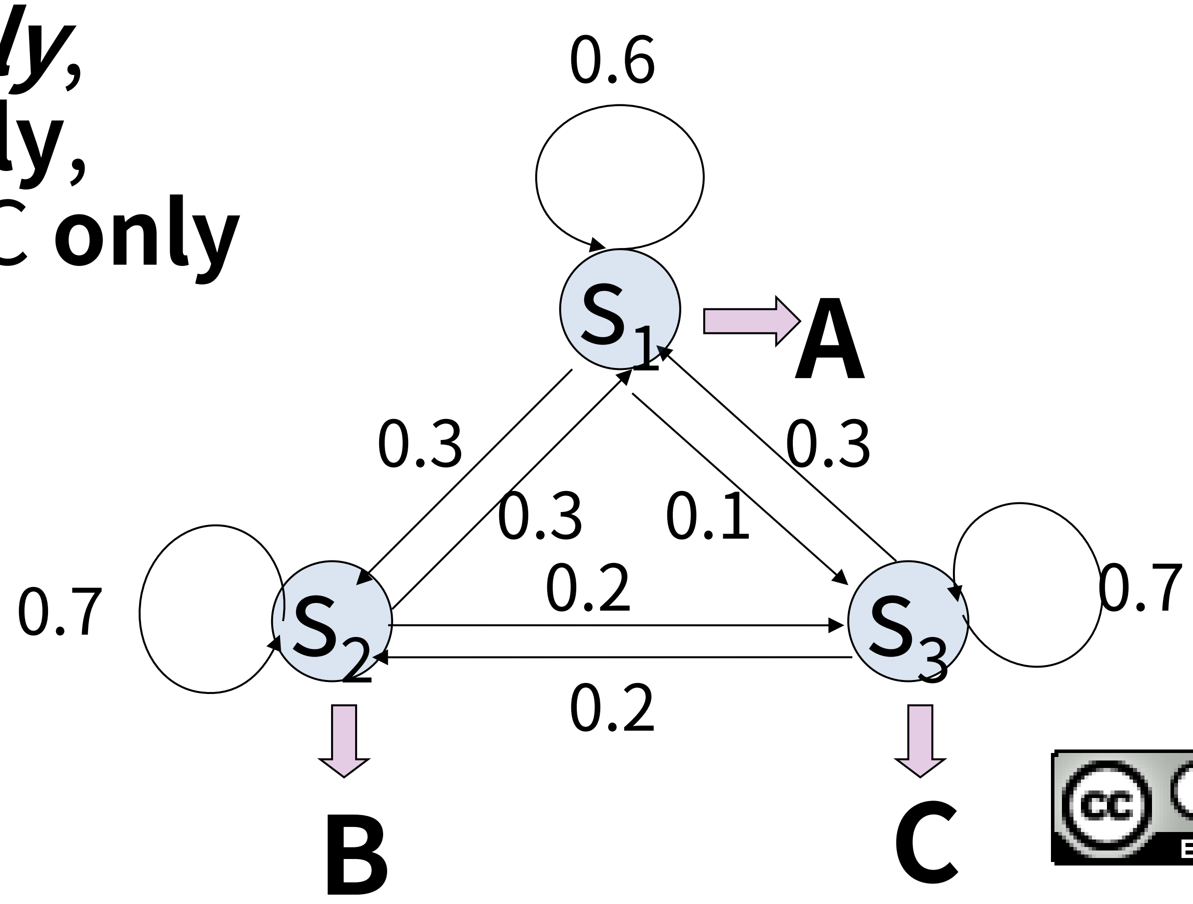
- **Markov Model (Markov Chain)**
  - First-order Markov chain of N states is a triplet (S,**A**,π)
    - S is a set of *N* states
    - **A** is the *N*⊠*N* matrix of state transition probabilities
      $P(q_t=j|q_{t-1}=i, q_{t-2}=k, \ldots\ldots)=P(q_t=j|q_{t-1}=$
    - π is the vector of initial state probal
      $\pi_j=P(q_0=j)$
  - The output for any given state is an observable event (deterministic)
  - The output of the process is a sequ
    of observable events



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

Markov Model

- **An example : 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**



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

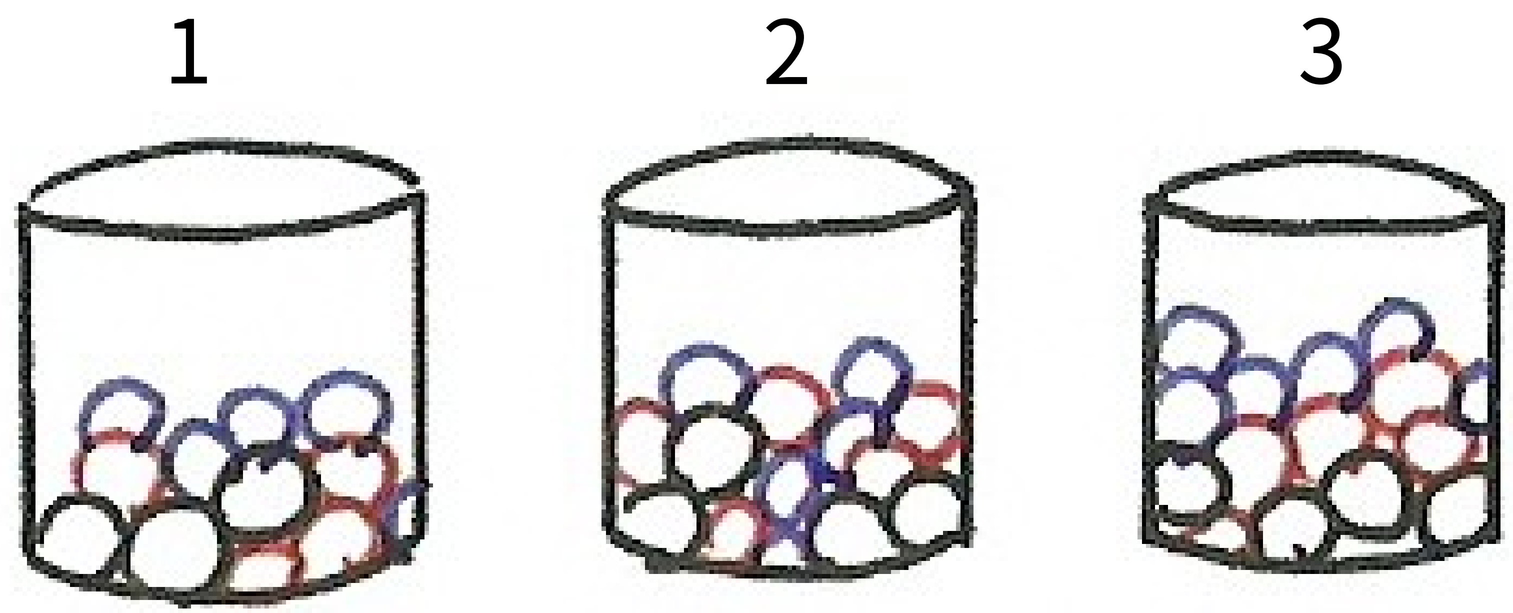
$$\pi = \begin{bmatrix} 0.4 & 0.5 & 0.1 \end{bmatrix}$$

  - Given a sequence of observed symbols **O**={CABBCABC}, the **only one** corresponding state sequence is $\{S_3 S_1 S_2 S_2 S_3 S_1 S_2 S_3\}$, and the corresponding probability is
    $P(\mathbf{O}|\boldsymbol{\lambda})=P(q_0=S_3)\ P(S_1/S_3)P(S_2/S_1)P(S_2/S_2)P(S_3/S_2)P(S_1/S_3)P(S_2/S_1)P(S_3/S_2)$
    $=0.1\boxtimes 0.3\boxtimes 0.3\boxtimes 0.7\boxtimes 0.2\boxtimes 0.3\boxtimes 0.3\boxtimes 0.2=0.00002268$

Hidden Markov Model

- **HMM, an extended version of Markov Model**
  - The observation is **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 never know which state sequence generates it
- **Elements of an HMM {S,A,B,π}**
  - S is a set of *N* states
  - **A** is the *N*⊠*N* matrix of state transition probabilities
  - 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

Simplified HMM



RGBGGBBGRRR……

Hidden Markov Model

- **Two types of HMM's according to the observation functions**

  Discrete and finite observations :

  - The observations that all distinct states generate are finite in number

    $\mathbf{V}=\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \ldots\ldots, \mathbf{v}_M\}, \mathbf{v}_k \in \boldsymbol{R}^D$

  - the set of observation probability distributions B={$b_j(\mathbf{v}_k)$} is defined as $b_j(\mathbf{v}_k)=P(\mathbf{o}_t=\mathbf{v}_k | \boldsymbol{q}_t=j), 1 \leq k \leq M, 1 \leq j \leq N$

    $\mathbf{o}_t$: *observation at time t,* $\boldsymbol{q}_t$: *state at time t*

    ☒ *for state j,* $b_j(\mathbf{v}_k)$ consists of ***only M probability values***

  Continuous and infinite observations :

  - The observations that all distinct states generate are infinite and continuous, $\mathbf{V}=\{\mathbf{v} | \mathbf{v} \in \boldsymbol{R}^D\}$

  - the set of observation probability distributions B={$b_j(\mathbf{v})$} is defined as $b_j(\mathbf{v})=\mathrm{P}(\mathbf{o}_t=\mathbf{v} | \boldsymbol{q}_t=j), 1 \leq j \leq N$

    ☒ $b_j(\mathbf{v})$ is a ***continuous probability density function and is often assumed to be a mixture of Gaussian distributions***

$$b_j(\mathbf{v})=\sum_{k=1}^{M} c_{jk}\left(\frac{1}{\left(\sqrt{2\pi}\right)^D \left|\Sigma_{jk}\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left(\left(\mathbf{v}-\boldsymbol{\mu}_{jk}\right)^t \Sigma_{jk}^{-1}\left(\mathbf{v}-\boldsymbol{\mu}_{jk}\right)\right)\right)\right)=\sum_{k=1}^{M} c_{jk} b_{jk}(V)$$

Hidden Markov Model

- **An example : a 3-state discrete HMM *λ***

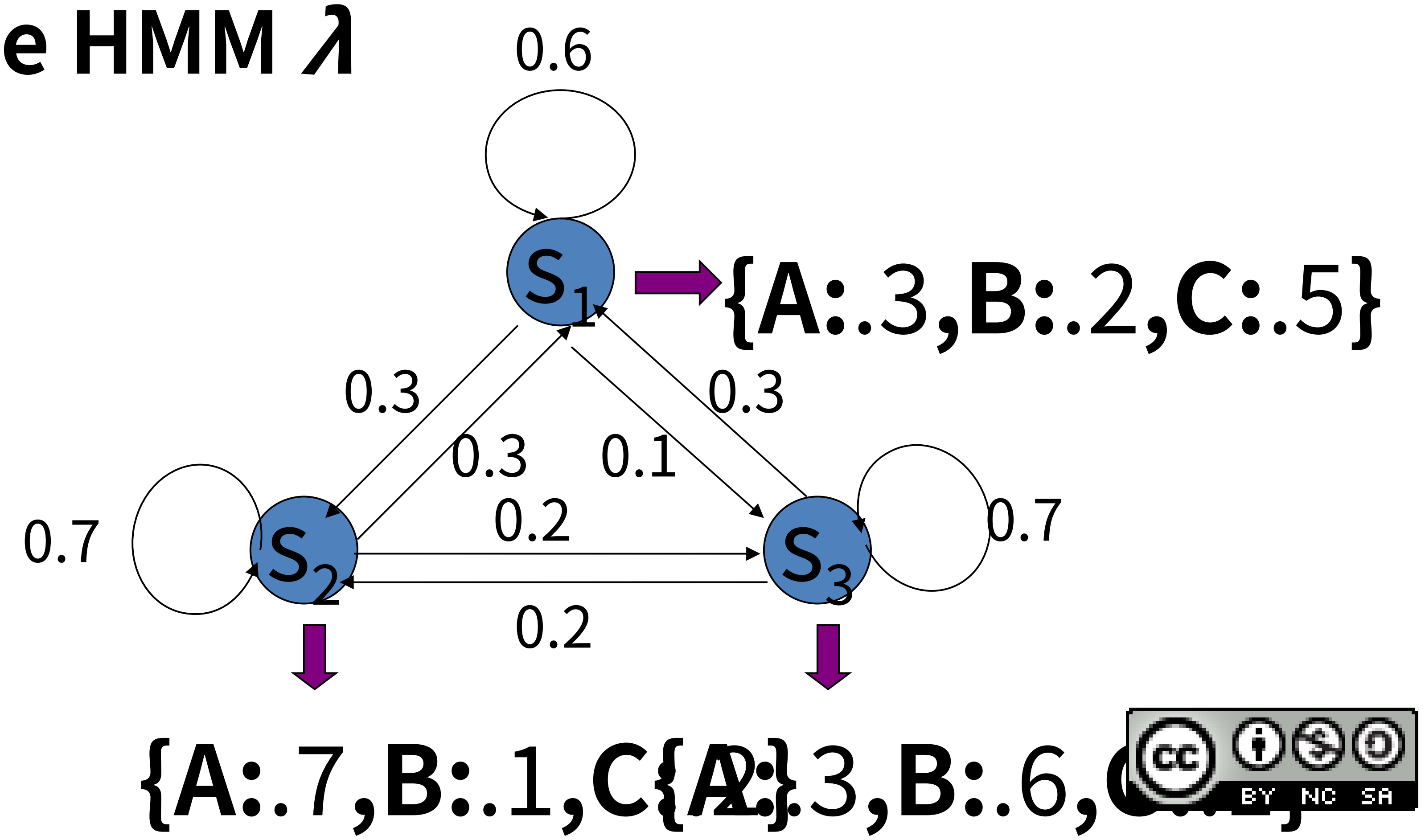
$$\mathbf{A} = \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.1 & 0.7 & 0.2 \\ 0.3 & 0.2 & 0.5 \end{bmatrix}$$

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

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

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

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



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

$$P(\bar{\mathbf{O}}|\lambda) = \sum_{i=1}^{27} P(\bar{\mathbf{O}}, \mathbf{q}_i|\lambda) = \sum_{i=1}^{27} P(\bar{\mathbf{O}}|\mathbf{q}_i, \lambda) P(\mathbf{q}_i|\lambda), \quad \mathbf{q}_i: \text{state sequence}$$

$$e.g. \text{ when } \mathbf{q}_i = \{S_2 S_2 S_3\}, P(\bar{\mathbf{O}}|\mathbf{q}_i, \lambda) = P(\mathbf{A}|S_2)P(\mathbf{B}|S_2)P(\mathbf{C}|S_3) = 0.7 * 0.1 * 0.1 = 0.007$$

$$P(\mathbf{q}_i|\lambda) = P(q_0 = S_2)P(S_2|S_2)P(S_3|S_2) = 0.5 * 0.7 * 0.2 = 0.07$$

Hidden Markov Model

- **Three Basic Problems for HMMs**

  Given an observation sequence $O=(o_1,o_2,\ldots\ldots,o_T)$**, and an HMM** $\lambda=(A,B,\pi)$

  - Problem *1*:

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

    ☒ *Evaluation problem*

  - Problem *2*:

    How to choose an optimal state sequence $\mathbf{q}=(q_1,q_2,\ldots\ldots, q_T)$?

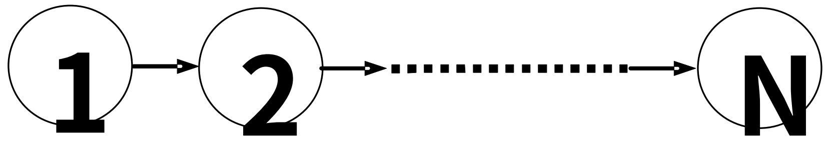
    ☒ *Decoding Problem*

  - Problem *3*:

    Given some observations O for the HMM λ, how to adjust the model parameter $\lambda=(A,B,\pi)$ to maximize $P(\mathbf{O}|\lambda)$?

    ☒ *Learning /Training Problem*

Basic Problem 1 for HMM



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

$\bar{O} = o_1 o_2 o_3 \cdots\cdots o_t \cdots\cdots o_T$ observation sequence

$\bar{q} = q_1 q_2 q_3 \cdots\cdots q_t \cdots\cdots q_T$ state sequence

☒**Problem 1:** Given λ and $\bar{O}$,

find $P(\bar{O}|\lambda)$ = Prob[observing $\bar{O}$ given λ]

☒**Direct Evaluation: considering all possible state sequence $\bar{q}$**

$$P(\bar{O}|\lambda) = \sum_{\text{all } \bar{q}} P(\bar{O},\bar{q}|\lambda) = \sum_{\text{all } \bar{q}} P(\bar{O}|\bar{q},\lambda)P(\bar{q}|\lambda)$$

$$P(\bar{O}|\bar{q},\lambda)$$

$$\Uparrow$$

$$P(\bar{O}|\lambda) = \sum_{\text{all } \bar{q}} \Big([b_{q_1}(o_1) \bullet b_{q_2}(o_2) \bullet \cdots\cdots b_{q_T}(o_T)] \bullet [\pi_{q_1} \bullet a_{q_1 q_2} \bullet a_{q_2 q_3} \bullet \cdots\cdots a_{q_{T-1} q_T}]\Big)$$

$$\Downarrow$$

$$P(\bar{q}|\lambda)$$

total number of different $\bar{q}$: N^T

huge computation requirements

Basic Problem 1 for HMM

- **Forward Algorithm: defining a forward variable $\alpha_t(i)$**

$$\alpha_t(i) = P(o_1 o_2 \cdots\cdots o_t\, ,\, q_t = i|\lambda)$$

$$=\text{Prob}[\text{observing } o_1 o_2 \cdots o_t\, ,\ \text{state } i \text{ at time } t|\lambda]$$

- Initialization

$$\alpha_1(i) = \pi_i b_i(o_1)\ ,\quad 1 \le i \le N$$

- Induction

$$\alpha_{t+1}(j) = [\sum_{i=1}^{N} \alpha_t(i) a_{ij}]\, b_j(o_{t+1})$$

$$1 \le j \le N$$
$$1 \le t \le T-1$$

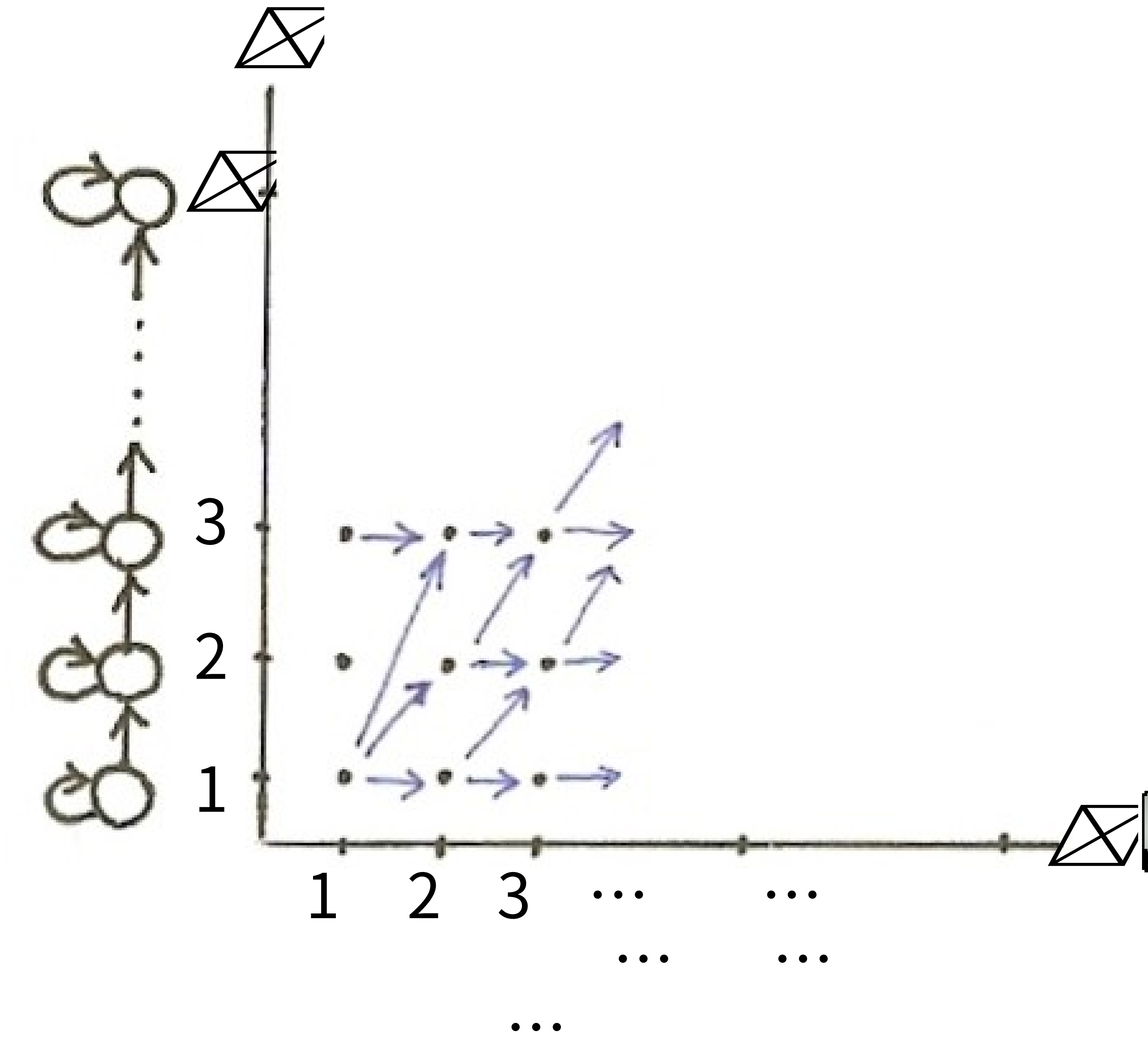
- Termination

$$P(\overline{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

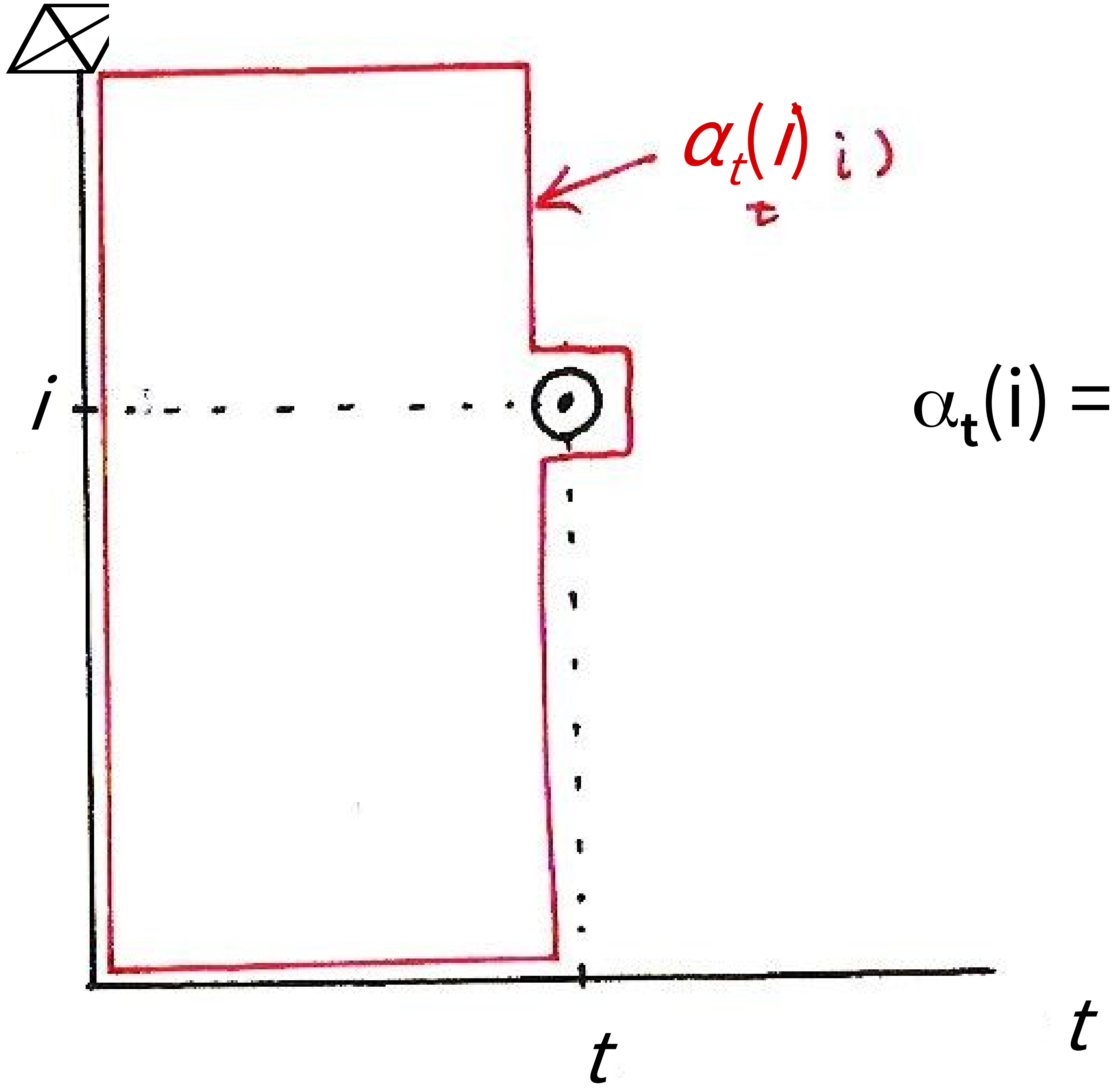
See Fig. 6.5 of Rabiner and Juang

- All state sequences, regardless of how long previously, merge to the N state at each time instant t

Basic Problem 1

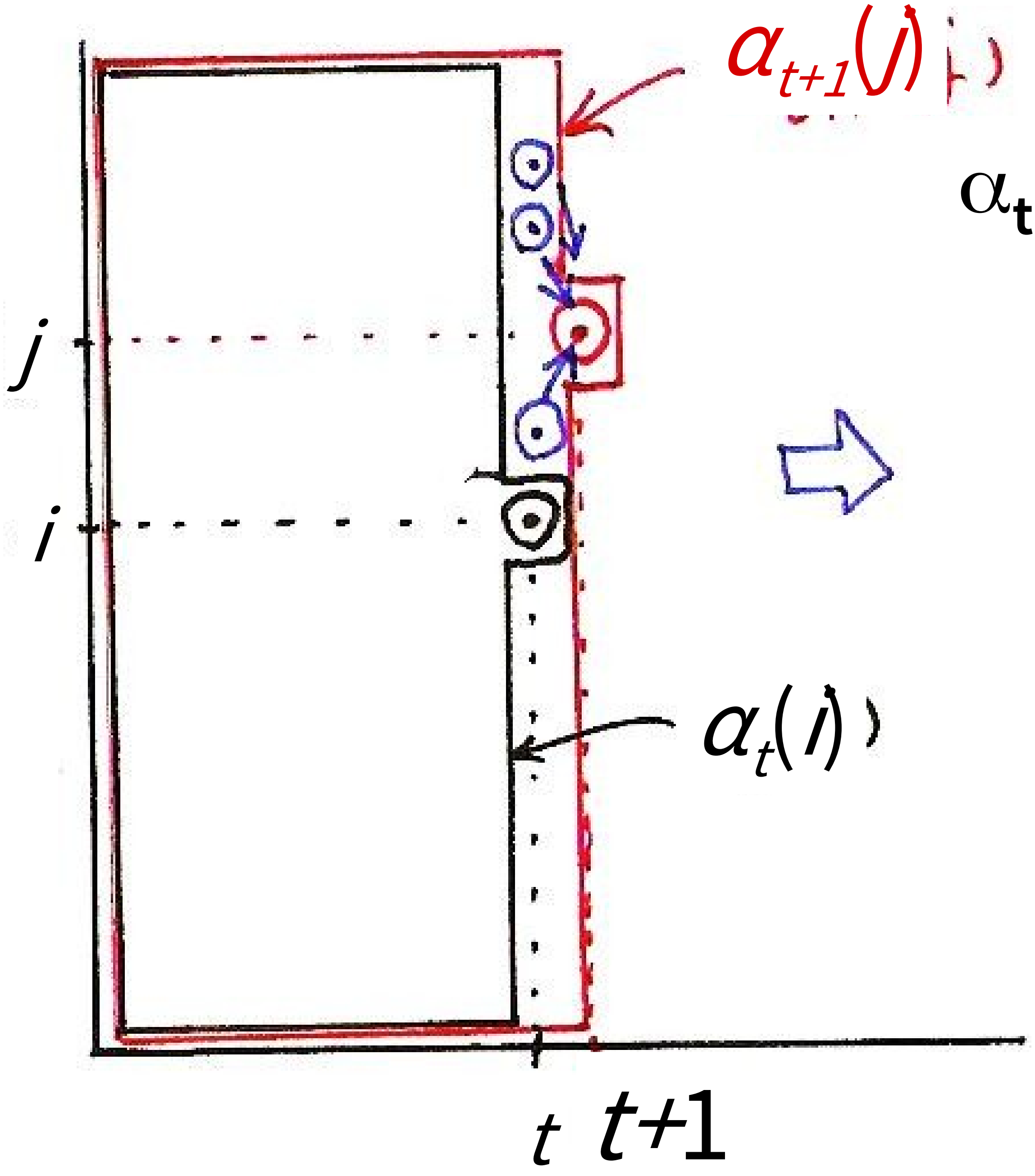


Basic Problem 1



$$\alpha_t(i) = P(o_1o_2......o_t, q_t = i|\lambda)$$

Basic Problem 1



$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(o_{t+1})$$

$$1 \le j \le N$$

$$1 \le t \le T\text{-}1$$

Forward Algorithm

Basic Problem 2 for HMM

- **Problem 2:** Given λ and $\overline{O} = o_1 o_2 \cdots o_T$, find a best state sequence $\overline{q} = q_1 q_2 \cdots q_T$
- **Backward Algorithm :** defining a backward variable $\beta_t(i)$

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \cdots, o_T \,|\, q_t = i, \lambda)$$

$$= \text{Prob}[\text{observing } o_{t+1}, o_{t+2}, \cdots, o_T \,|\, \text{state } i \text{ at time } t, \lambda]$$

- Initialization

$$\beta_T(i) = 1, \quad 1 \le i \le N \qquad \left(\beta_{T-1}(i) = \sum_{j=1}^{N} a_{ij}\, b_j(o_T) \right)$$

- Induction

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}\, b_j(o_{t+1}) \beta_{t+1}(j)$$

$$t = T-1, T-2, \cdots, 2, 1, \qquad 1 \le i \le N$$

See Fig. 6.6 of Rabiner and Juang

- **Combining Forward/Backward Variables**

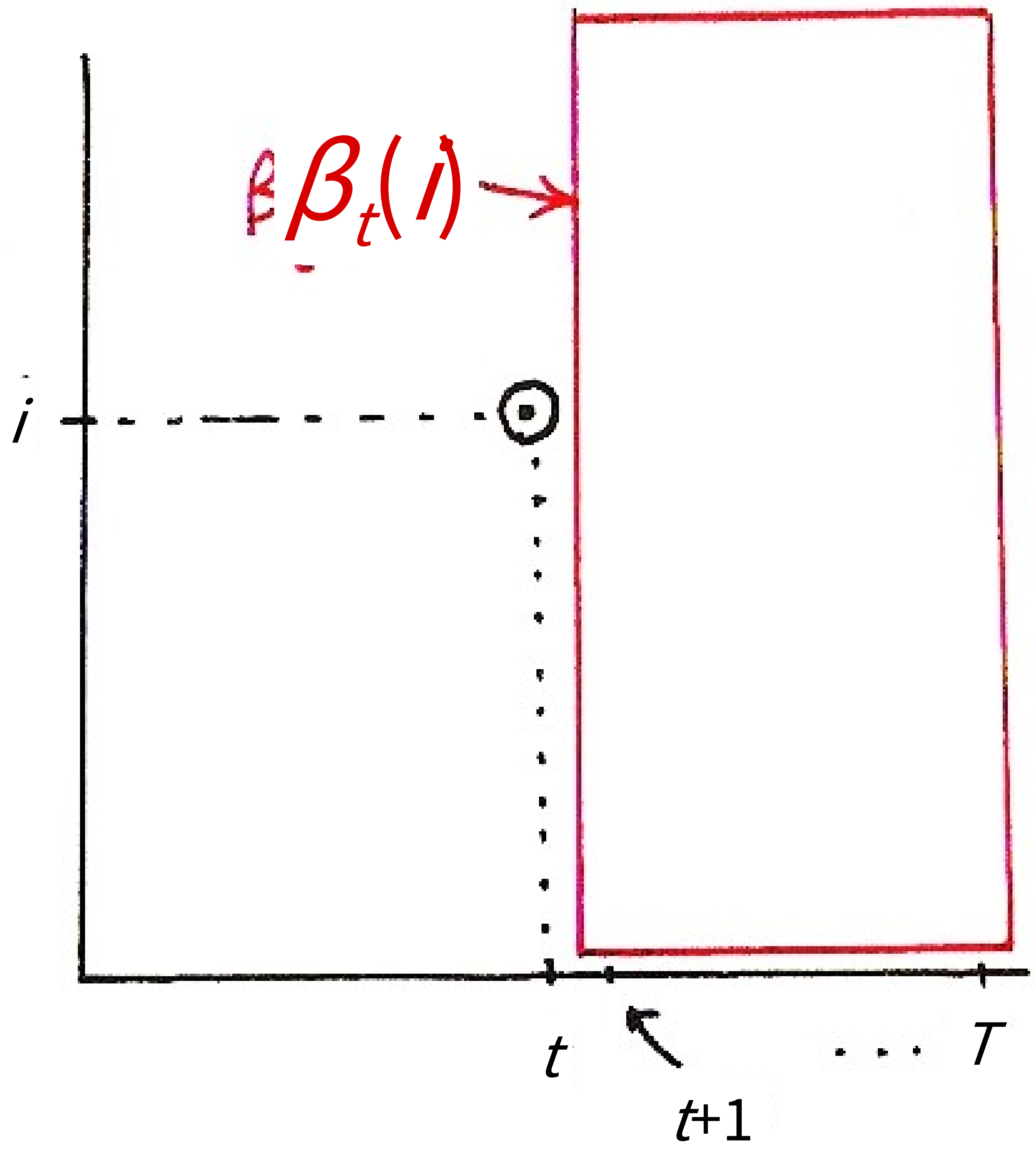
$$P(\overline{O}, q_t = i \,|\, \lambda)$$

$$= \text{Prob}\,[\text{observing } o_1, o_2, \cdots, o_t, \cdots, o_T, q_t = i \,|\, \lambda\,]$$

$$= \alpha_t(i)\beta_t(i)$$

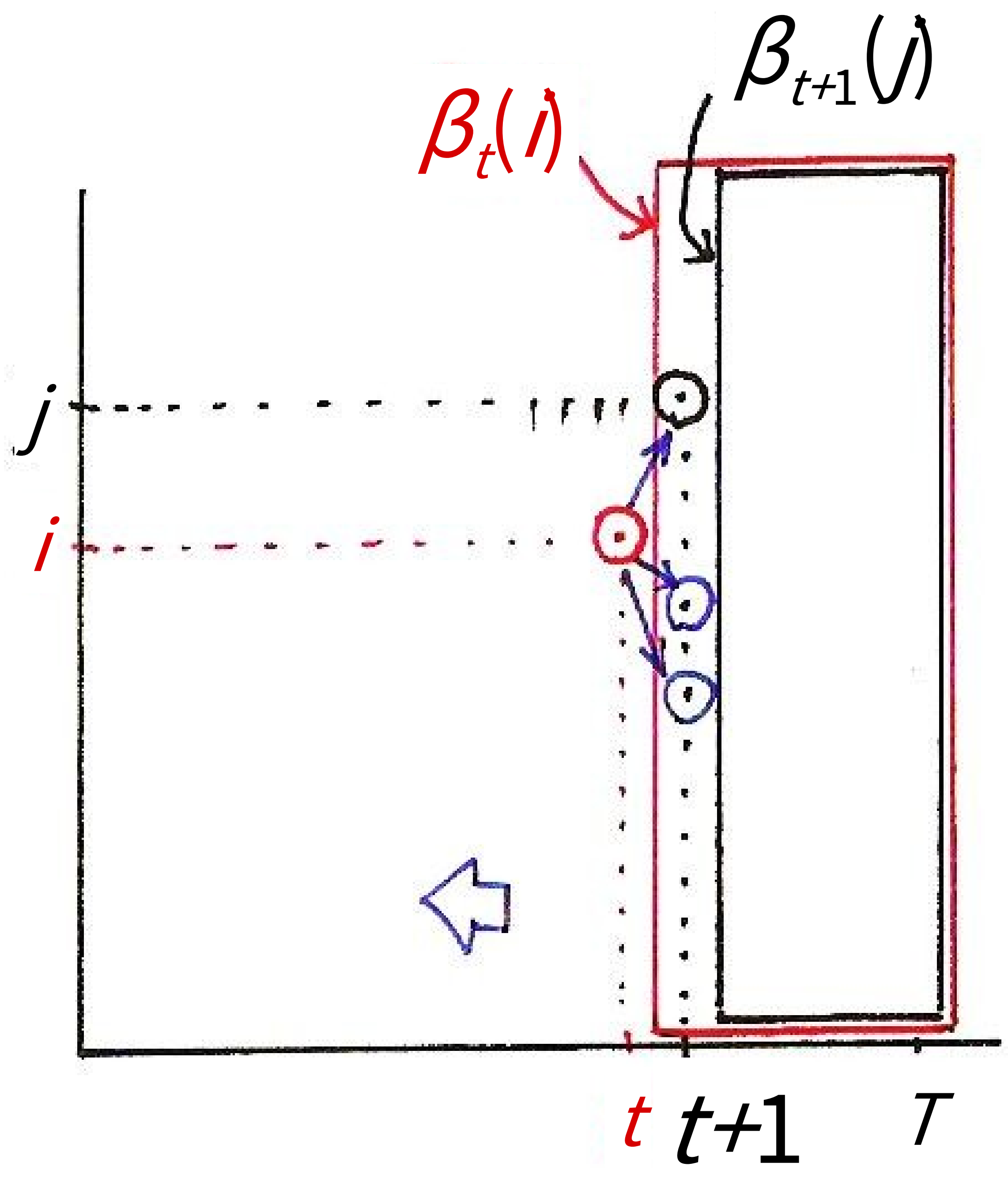
$$P(\overline{O}|\lambda) = \sum_{i=1}^{N} P(\overline{O}, q_t = i \,|\, \lambda) = \sum_{i=1}^{N} [\alpha_t(i)\beta_t(i)]$$

Basic Problem 2



$\beta_t(i) = P(o_{t+1}, o_{t+2}, \ldots, o_T | q_t = i, \lambda)$

Basic Problem 2

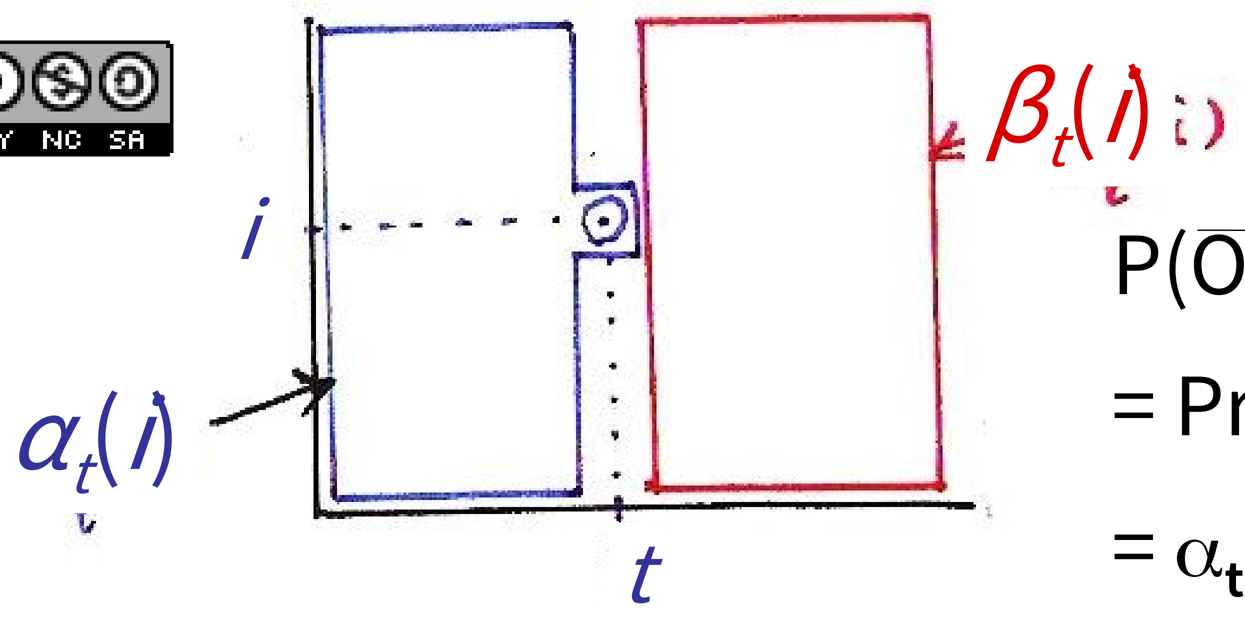


$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}\, b_j(o_{t+1})\beta_{t+1}(j)$$

$t = T-1, T-2, \cdots, 2, 1, \quad 1 \le i \le N$

Backward Algorithm

Basic Problem 2



$P(\overline{O}, q_t = i \mid \lambda)$

$= \text{Prob [observing } o_1, o_2, \ldots, o_t, \ldots, o_T, q_t = i \mid \lambda\,]$

$= \alpha_t(i)\beta_t(i)$

$$\boxtimes(\boxtimes,\boxtimes,\boxtimes)=\boxtimes(\boxtimes,\boxtimes)\,\boxtimes(\boxtimes\,|\,\boxtimes,\boxtimes)$$

$$// \qquad\qquad // \qquad\qquad // \qquad\qquad (\boxtimes\perp\boxtimes)$$

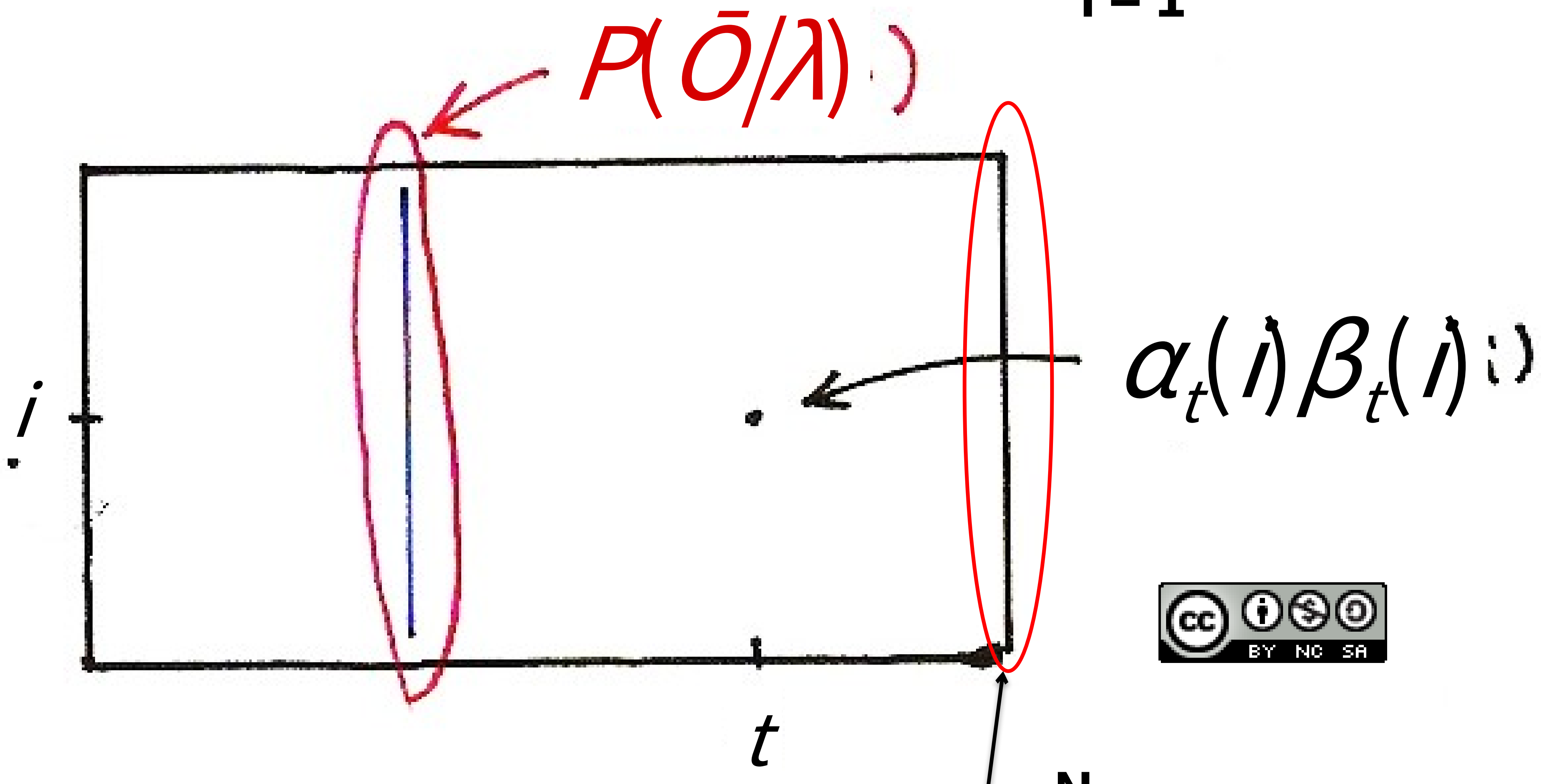
$$\boxtimes(\overline{\boxtimes},\boxtimes_{\boxtimes}=\boxtimes\,|\,\boxtimes)\,\boxtimes_{\boxtimes}(\boxtimes) \qquad \boxtimes(\boxtimes\,|\,\boxtimes)$$

$$//$$

$$\boxtimes_{\boxtimes}(\boxtimes)$$

Basic Problem 2

$$P(\bar{O}|\lambda) = \sum_{i=1}^{N} P(\bar{O}, q_t = i\,|\lambda) = \sum_{i=1}^{N} [\alpha_t(i)\beta_t(i)]$$



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

Basic Problem 2 for HMM

- **Approach 1 – Choosing state q_t individually as the most likely state at time t**

  - Define a new variable $\gamma_t(i) = P(q_t = i \mid \bar{O}, \lambda)$

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

  - Solution

  $$q_t^* = \arg\max_{1 \le i \le N} [\gamma_t(i)], \quad 1 \le t \le T$$

  in fact

  $$q_t^* = \arg\max_{1 \le i \le N} [P(\bar{O}, q_t = i|\lambda)]$$
  $$= \arg\max_{1 \le i \le N} [\alpha_t(i)\beta_t(i)]$$

  - Problem

  maximizing the probability at each time t individually

  $\bar{q}^* = q_1^* q_2^* \ldots q_T^*$ may not be a valid sequence

  (e.g. $a_{q_t^* q_{t+1}^*} = 0$)

Basic Problem 2 for HMM

- **Approach 2 – Viterbi Algorithm - finding the single best sequence $\bar{q}^* = q_1^* q_2^* \cdots q_T^*$**
  - Define a new variable $\delta_t(i)$

    $$\delta_t(i) = \max_{q_1,q_2,\cdots q_{t-1}} P[q_1,q_2,\cdots q_{t-1}, q_t = i, o_1,o_2,\cdots,o_t \mid \lambda]$$

    = the highest probability along a certain single path ending at state i at time t for the first t observations, given λ

  - Induction

    $$\delta_{t+1}(j) = \max_i [\delta_t(i)a_{ij}] \bullet b_j(o_{t+1})$$

  - Backtracking

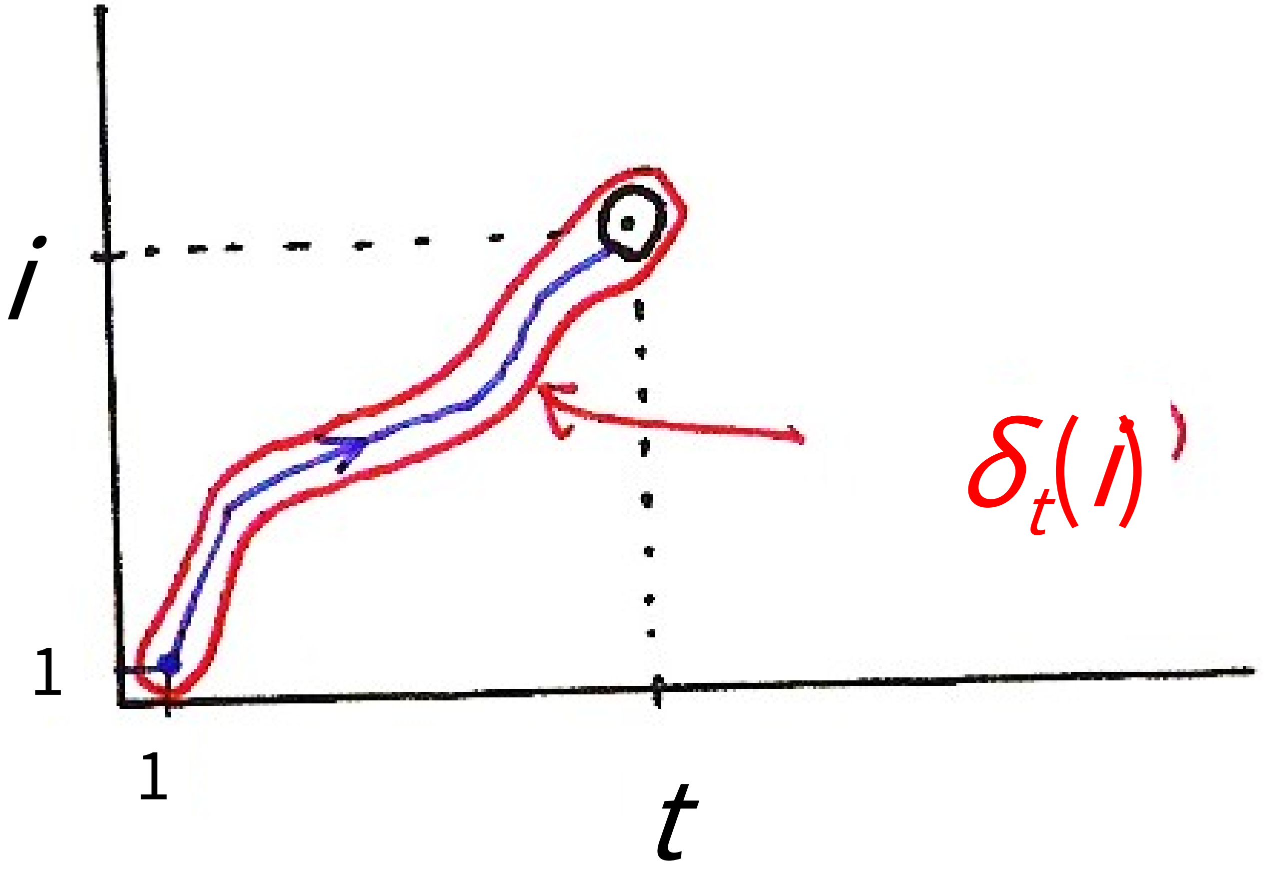
    $$\psi_{t+1}(j) = \arg\max_{1 \le i \le N} [\delta_t(i)a_{ij}]$$

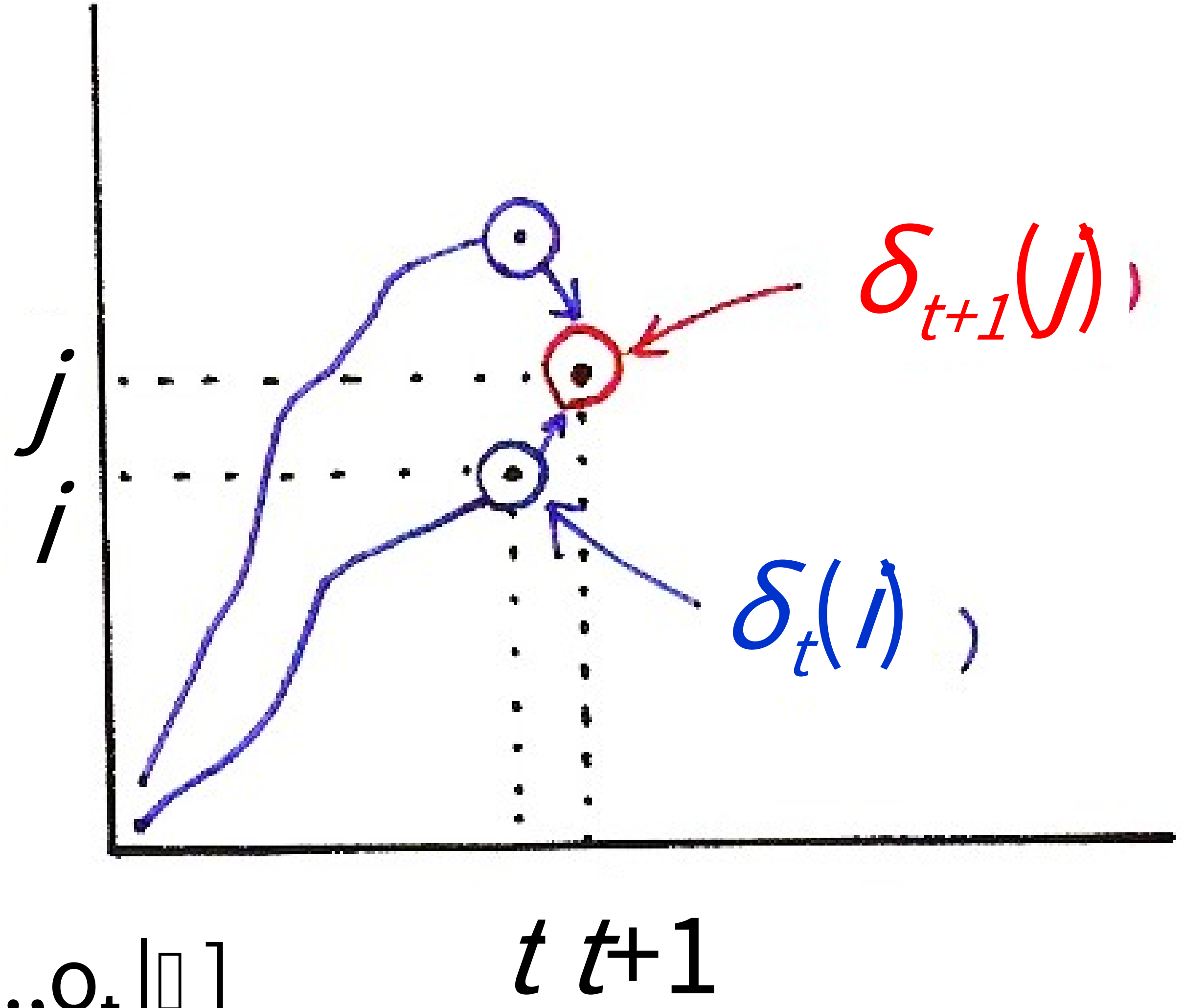
    the best previous state at t- 1 given at state j at time t

    keeping track of the best previous state for each j and t

Viterbi Algorithm

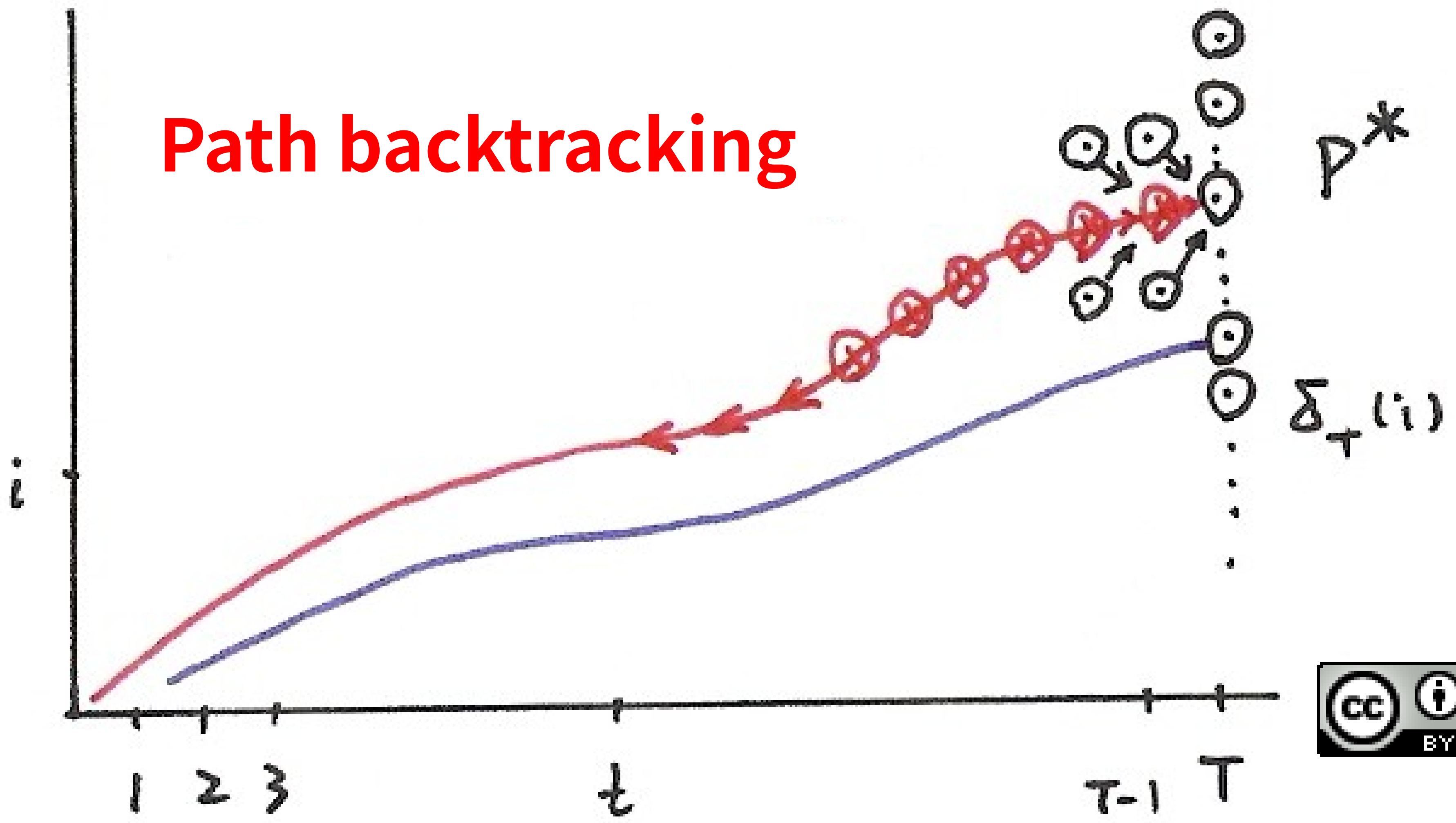
$$\delta_{t+1}(j) = \max_{i} [\delta_t(i)a_{ij}] \bullet b_j(o_{t+1})$$





$$\square_t(i) = \max_{q_1,q_2,\ldots q_{t-1}} P[q_1,q_2,\ldots q_{t-1}, q_t = i, o_1,o_2,\ldots,o_t|\square]$$

Viterbi Algorithm



Basic Problem 2 for HMM

- **Complete Procedure for Viterbi Algorithm**

  - Initialization

    $\delta_1(i) = \pi_i b_i(o_1)$, $1 \le i \le N$

  - Recursion

    $\delta_{t+1}(j) = \max\limits_{1 \le i \le N} [\delta_t(i) a_{ij}] \bullet b_j(o_t)$

    $2 \le t \le T, \quad 1 \le j \le N$

    $\psi_{t+1}(j) = \arg \max\limits_{1 \le i \le N} [\delta_t(i) a_{ij}]$

    $2 \le t \le T, \quad 1 \le j \le N$

  - Termination

    $P^* = \max\limits_{1 \le i \le N} [\delta_T(i)]$

    $q_T^* = \arg \max\limits_{1 \le i \le N} [\delta_T(i)]$

  - Path backtracking

    $q_t^* = \psi_{t+1}(q^*_{t+1})$, $t = T-1, t-2, \cdots..2, 1$

Basic Problem 2 for HMM

☒ **Application Example of Viterbi Algorithm**

- Isolated word recognition

$$\lambda_0 = (A_0, B_0, \pi_0)$$
$$\lambda_1 = (A_1, B_1, \pi_1)$$
$$\vdots$$
$$\lambda_n = (A_n, B_n, \pi_n)$$

observation

$$\bar{O} = (o_1, o_2, \ldots o_T)$$

$$k^* = \underset{1\le i\le n}{\arg\max} P[\bar{O}|\lambda_i] \approx \underset{1\le i\le n}{\arg\max} [P^*|\lambda_i]$$

⇧ ⇧

Basic Problem 1
Forward Algorithm
(for all paths)

Basic Problem 2
Viterbi Algorithm
(for a single best path)

-The model with the highest probability for the most probab
usually also has the highest probability for all possible path

Basic Problem 3 for HMM

- **Problem 3:** Give $\overline{O}$ and an initial model $\lambda=(A,B,\pi)$, adjust λ to maximize $P(\overline{O}|\lambda)$
  - Baum-Welch Algorithm (Forward-backward Algorithm)
  - Define a new variable

$$\varepsilon_t(i,j) = P(q_t = i, q_{t+1} = j \mid \overline{O}, \lambda)$$

$$= \frac{\alpha_t(i)\, a_{ij}\, b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}[\alpha_t(i)a_{ij}\, b_j(o_{t+1})\beta_{t+1}(j)]}$$

$$= \frac{Prob[\overline{O}, q_t = i, q_{t+1} = j|\lambda]}{P(\overline{O}|\lambda)}$$

See Fig. 6.7 of Rabiner and Juang

  - Recall $\gamma_t(i) = P(q_t = i \mid \overline{O}, \lambda)$

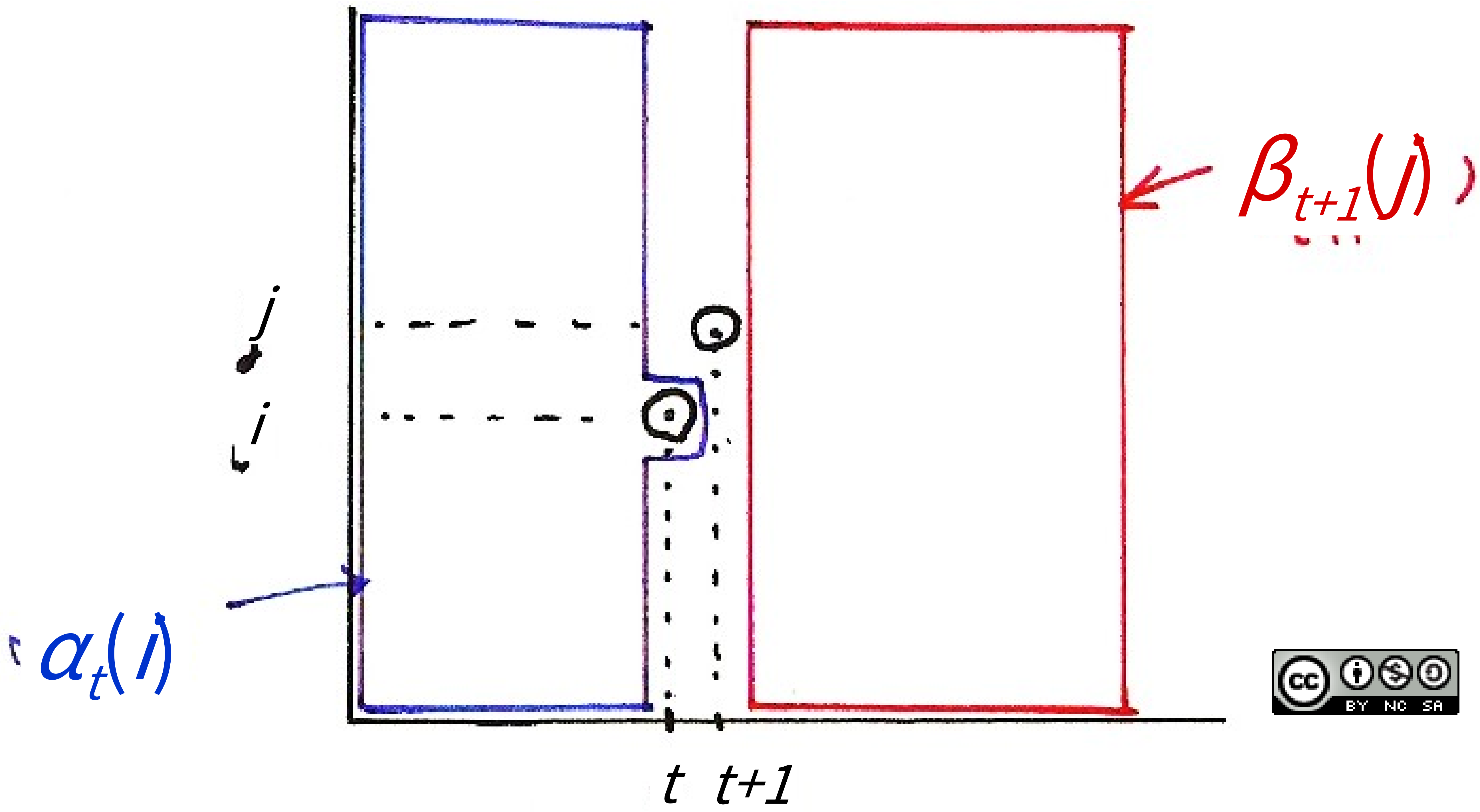
$\sum_{t=1}^{T-1}\gamma_t(i)$ = expected number of times that state i is visited in $\overline{O}$ from t = 1 to t = T-1

= expected number of transitions from state i in $\overline{O}$

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

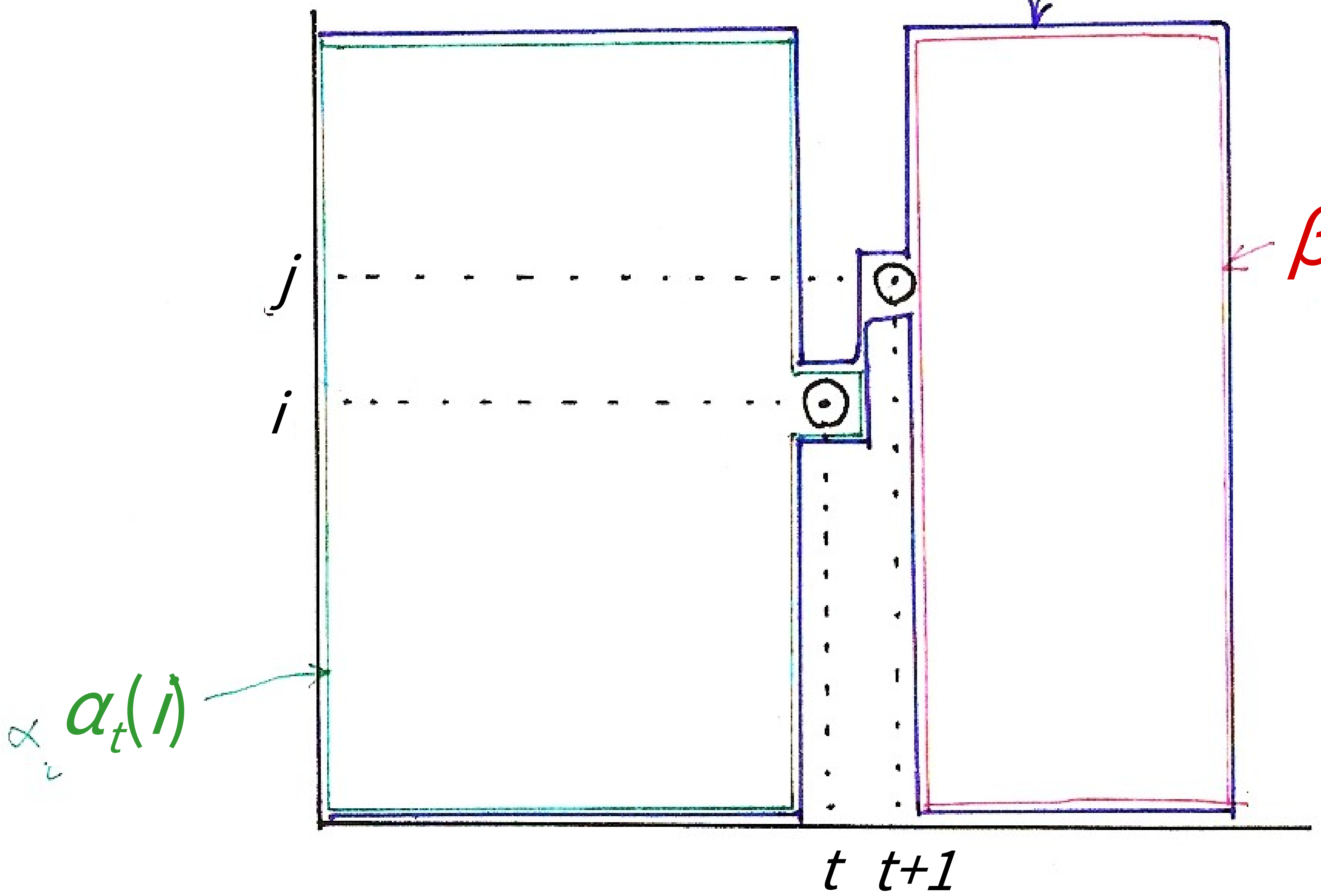
Basic Problem 3

$\alpha_t(i)\, a_{ij}\, b_j(o_{t+1})\beta_{t+1}(j)$



Basic Problem 3

$\alpha_t(i)\, a_{ij} b_j(o_{t+1})\, \beta_{t+1}(j)$



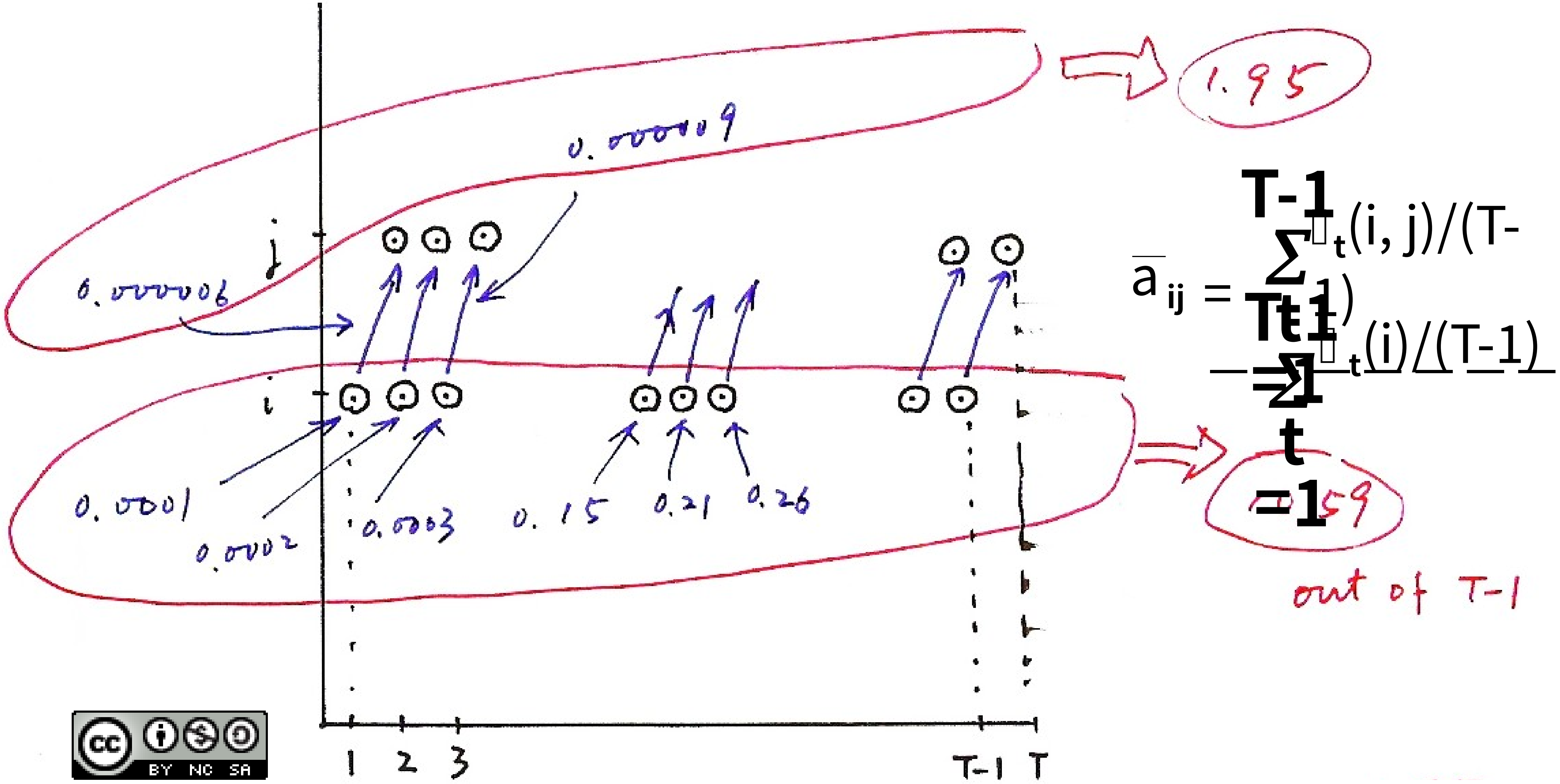
Basic Problem 3

[illegible]

[illegible]

[illegible]

Basic Problem 3



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

$$\bar{a}_{ij} = \frac{1.95/69}{10.59/69}$$

Basic Problem 3 for HMM

- Results

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

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

$$\bar{b}_j(k) = \text{Prob}[o_t = v_k \mid q_t = j\,] = \frac{\sum_{\substack{t=1 \\ o_t = v_k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

(for discrete HMM)

- **Continuous Density HMM**

$$b_j(o) = \sum_{k=1}^{M} c_{jk} N(o; \mu_{jk}, U_{jk})$$

N( ): Multi-variate Gaussian

μ_{jk}: mean vector for the k-th mixture component

U_{jk}: covariance matrix for the k-th mixture component

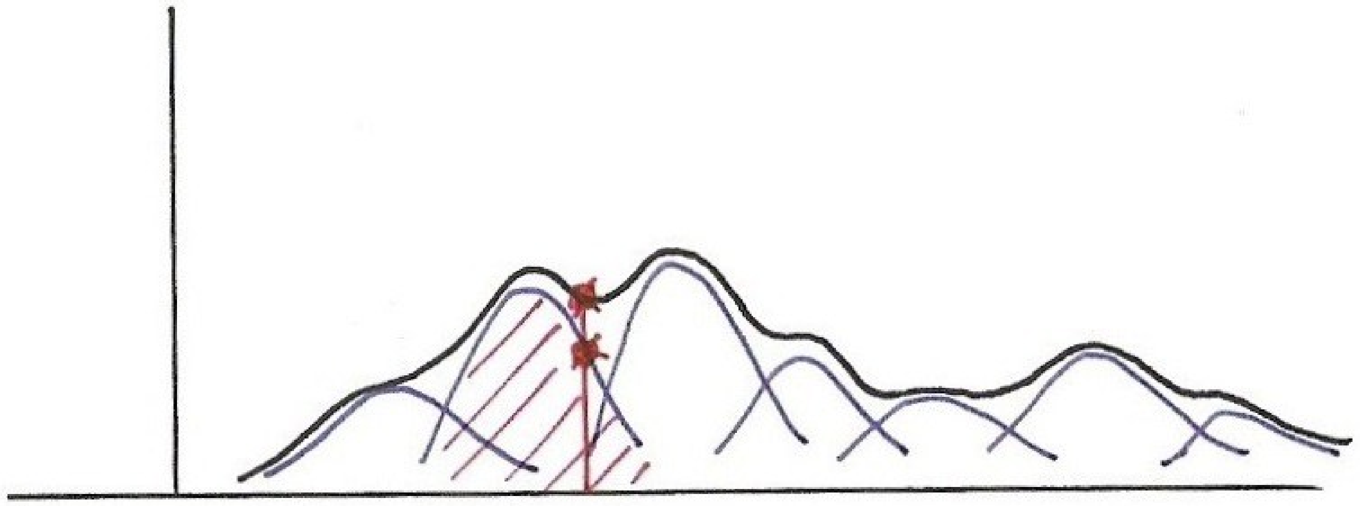
$\sum_{k=1}^{M} c_{jk} = 1$ for normalization

Basic Problem 3 for HMM

- **Continuous Density HMM**
  - Define a new variable

    $\gamma_t(j,k) = \gamma_t(j)$ but including the probability of o_t evaluated in the k-th mixture component out of all the mixture components

$$= \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \right] \left[\frac{c_{jk} N(o_t;\, \mu_{jk},\, U_{jk})}{\sum_{m=1}^{M} c_{jm} N(o_t;\, \mu_{jm},\, U_{jm})]} \right]$$



  - Results

$$\bar{c}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j,k)}$$

See Fig. 6.9 of Rabiner and Juang

Basic Problem 3 for HMM

- Continuous Density HMM

$$\bar{\mu}_{jk} = \frac{\sum_{t=1}^{T} [\gamma_t(j, k) \bullet o_t]}{\sum_{t=1}^{T} \gamma_t(j, k)}$$

$$\bar{U}_{jk} = \frac{\sum_{t=1}^{T} [\gamma_t(j, k)(o_t - \mu_{jk})\,(o_t - \mu_{jk})']}{\sum_{t=1}^{T} \gamma_t(j, k)}$$

- Iterative Procedure

$$\begin{tikzcd}
\lambda = (A, B, \pi) \arrow[rr] & & \bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi}) \arrow[ll, bend left=60] \\
& \bar{O} = o_1 o_2 \cdots o_T \arrow[ur] &
\end{tikzcd}$$

- It can be shown (by EM Theory (or EM Algorithm))

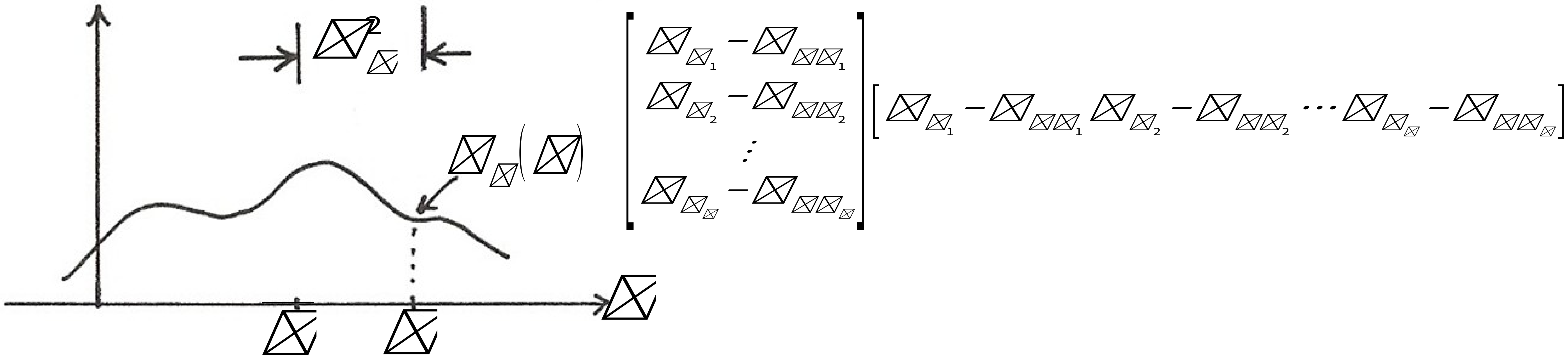
  $P(\bar{O}|\bar{\lambda}) \geq P(\bar{O}|\lambda)$ after each iteration

Basic Problem 3

$$\overline{\mu}_{jk} = \frac{\sum_{t=1}^{T}[\gamma_t(j,k) \cdot o_t]}{\sum_{t=1}^{T}\gamma_t(j,k)}$$

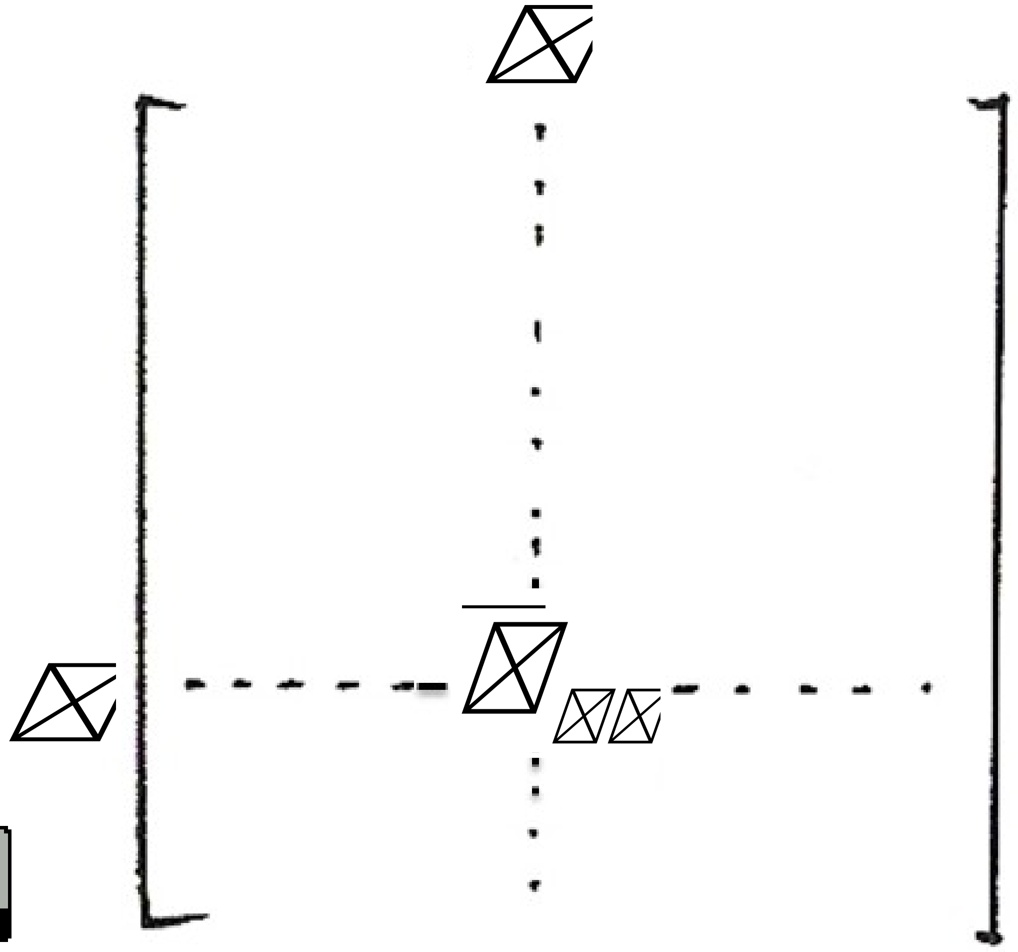
$$\overline{U}_{jk} = \frac{\sum_{t=1}^{T}[\gamma_t(j,k)\boxed{(o_t-\mu_{jk})(o_t-\mu_{jk})'}]}{\sum_{t=1}^{T}\gamma_t(j,k)}$$

prob. density function



Basic Problem 3

$$\bar{U} =$$



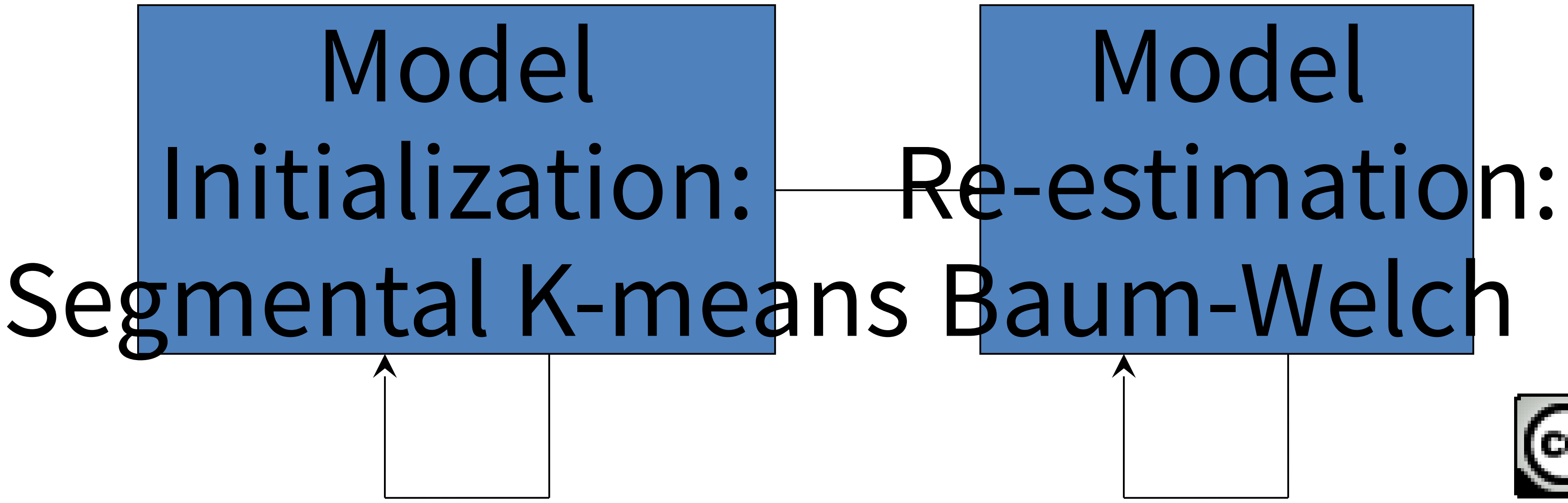
$$= E\left(\begin{bmatrix} x_1 - \bar{x}_1 \\ x_2 - \bar{x}_2 \\ \vdots \\ \vdots \end{bmatrix} [x_1 - \bar{x}_1, x_2 - \bar{x}_2, \cdots]\right)$$

$$\bar{u}_{lm} = E[(x_l - \bar{x}_l)(x_m - \bar{x}_m)]$$

Basic Problem 3 for HMM

- No closed-form solution, but approximated iteratively
- An initial model is needed-model initialization
- May converge to local optimal points rather than global optimal point

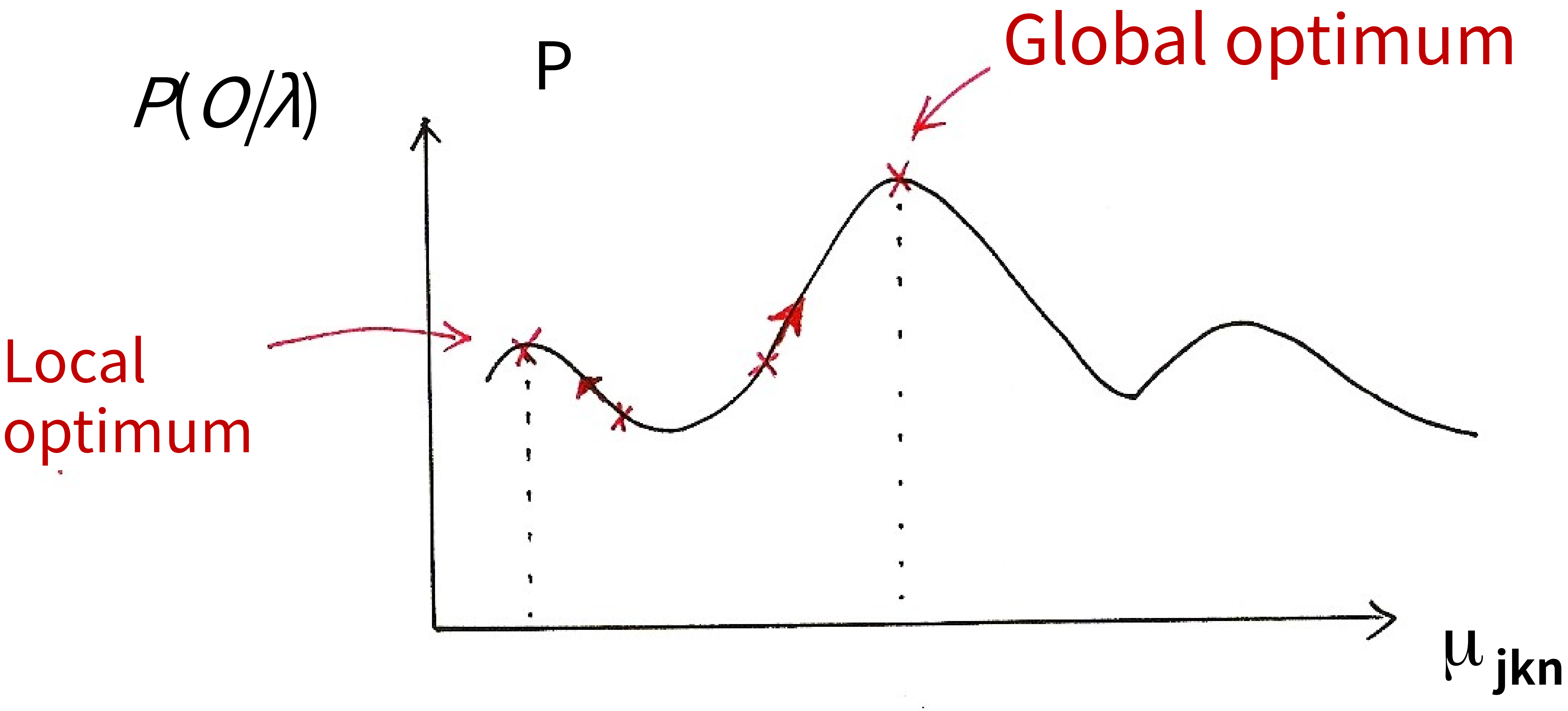
  - heavily depending on the initialization
- Model training



Model Initialization: Segmental K-means

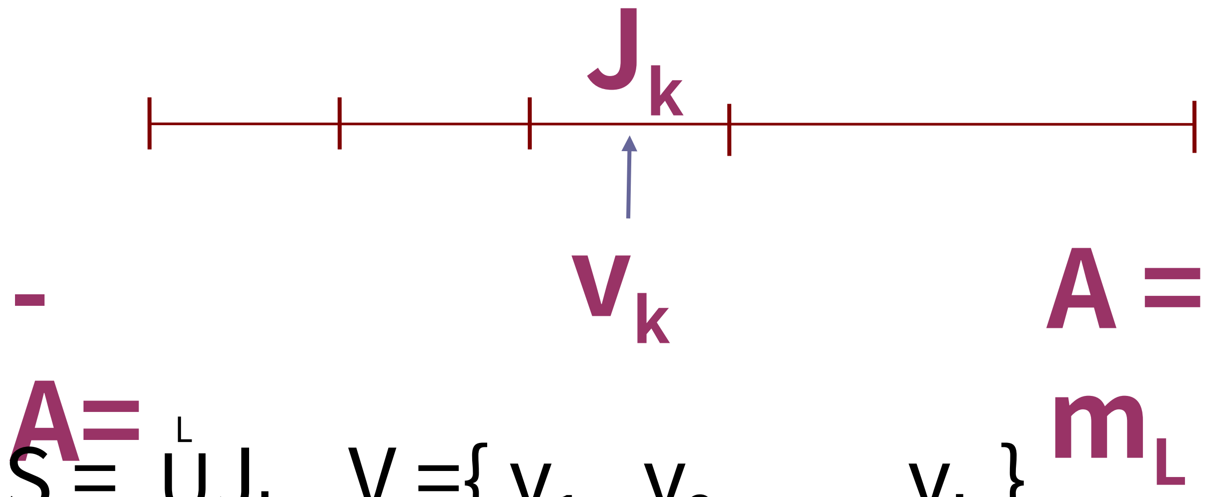
Model Re-estimation: Baum-Welch

Basic Problem 3



Vector Quantization (VQ)

- **An Efficient Approach for Data Compression**
  - replacing a set of real numbers by a finite number of bits
- **An Efficient Approach for Clustering Large Number of Sample Vectors**
  - grouping sample vectors into clusters, each represented by a single vector (codeword)
- **Scalar Quantization**
  - replacing a single real number by an R-bit pattern
  - a mapping relation



$S = \bigcup_{k=1}^{L} J_k$, $V = \{ v_1 , v_2 , \ldots, v_L \}$

$Q : S \rightarrow V$

$Q(x[n]) = v_k$ if $x[n] \in J_k$

$L = 2^R$

Each v_k represented by an R-bit pattern

–Quantization characteristics (codebook)

$\{ J_1 , J_2 , \ldots, J_L \}$ and $\{ v_1 , v_2 , \ldots, v_L \}$

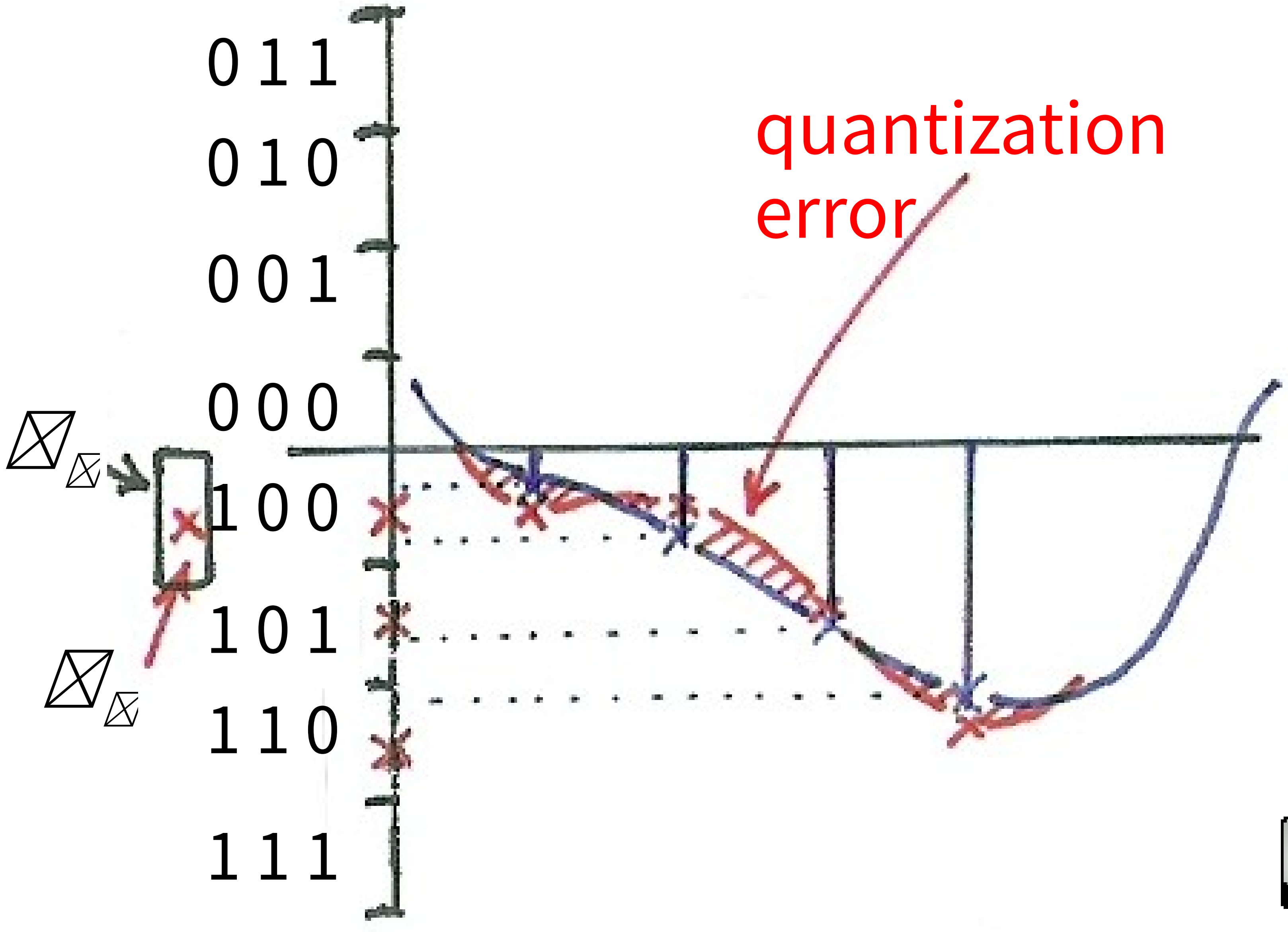
designed considering at least

1.error sensitivity

2.probability distribution of x[n]

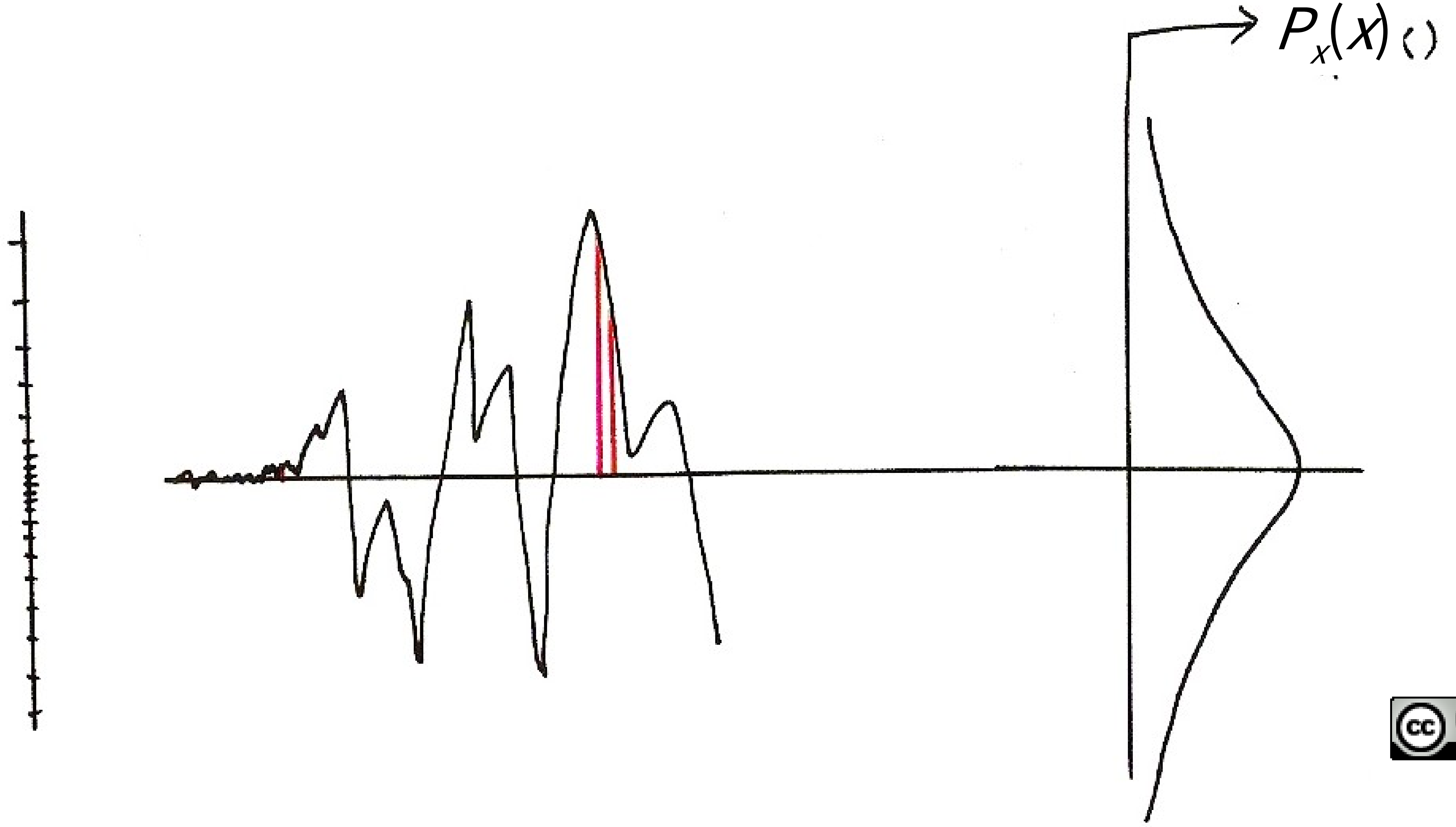
Vector Quantization

Scalar Quantization ： Pulse Coded Modulation (PCM)



100 100 101 110

Vector Quantization



Vector Quantization (VQ)

2-dim Vector Quantization (VQ)

Example:

$\bar{x}_n = (x[n] , x[n+1])$

$S = \{\bar{x}_n = (x[n] , x[n+1]) ; |x[n]| < A, |x[n+1]| < A\}$

- **VQ**
  - S divided into L 2-dim regions

    $\{ J_1 , J_2 , \ldots J_L \}$ $S = \bigcup_{k=1}^{L} J_k$

    each with a representative vector $\bar{v}_k \in J_k$, $V = \{ \bar{v}_1 , \bar{v}_2 , \ldots, \bar{v}_L \}$
  - $Q : S \Rightarrow V$

    $Q(\bar{x}_n) = \bar{v}_k$ if $\bar{x}_n \in J_k$

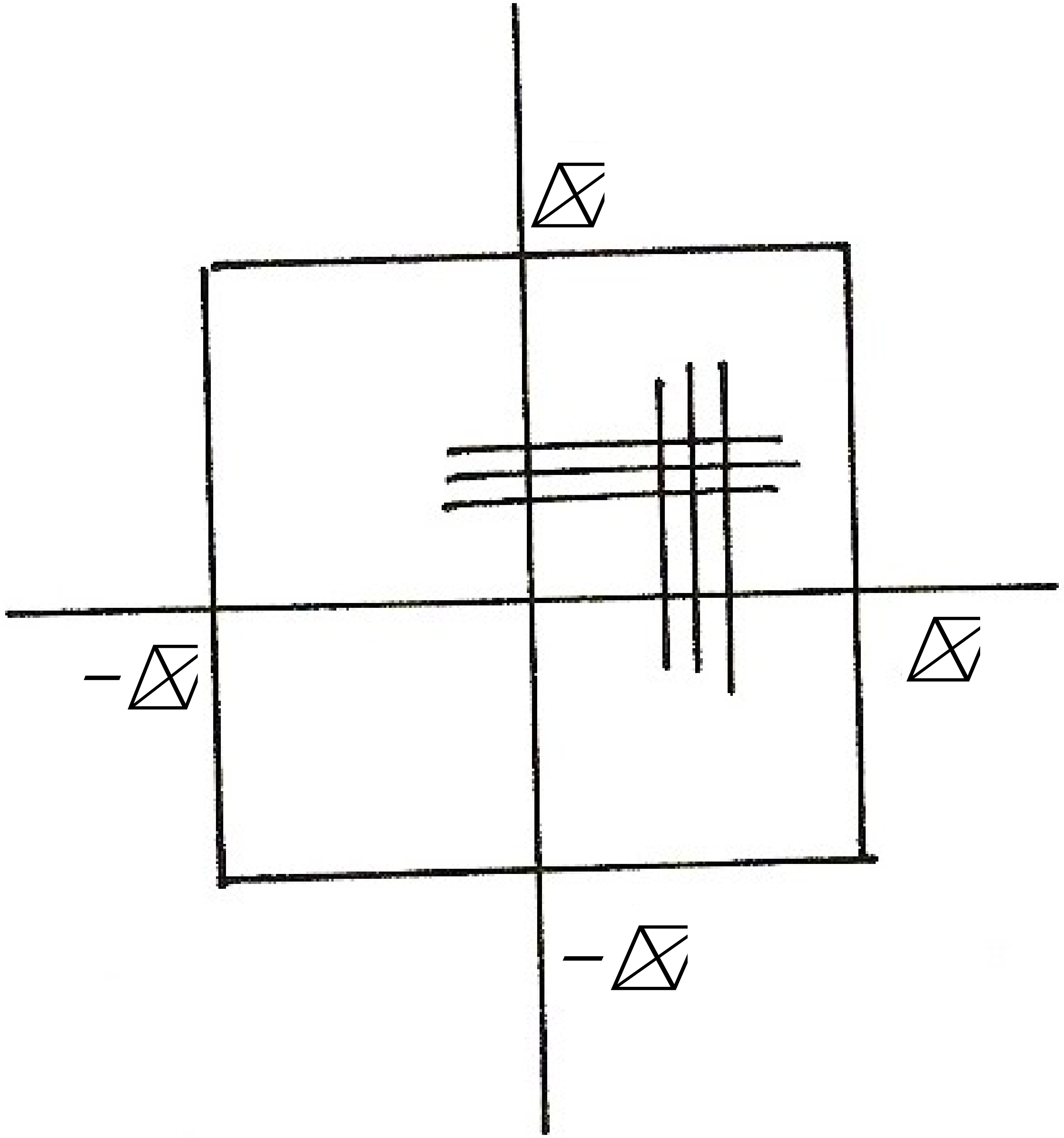
    $L = 2^R$

    each $\bar{v}_k$ represented by an R-bit pattern
  - Considerations

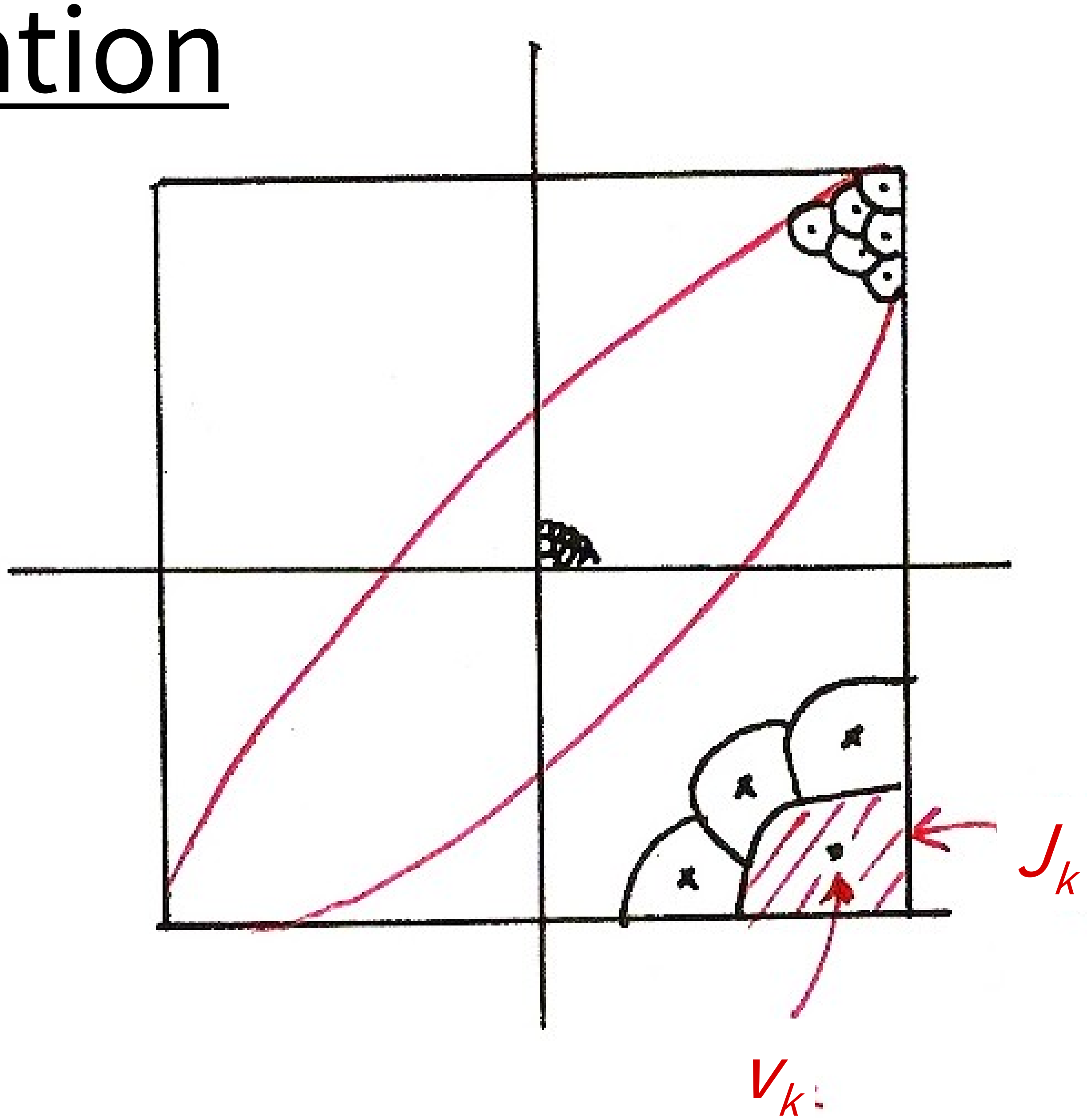
    1. error sensitivity may depend on x[n], x[n+1] jointly
    2. distribution of x[n] , x[n+1] may be correlated statistically
    3. more flexible choice of J_k
  - Quantization Characteristics (codebook)

    $\{ J_1 , J_2 , \ldots, J_L \}$ and $\{ \bar{v}_1 , \bar{v}_2 , \ldots, \bar{v}_L \}$

Vector Quantization



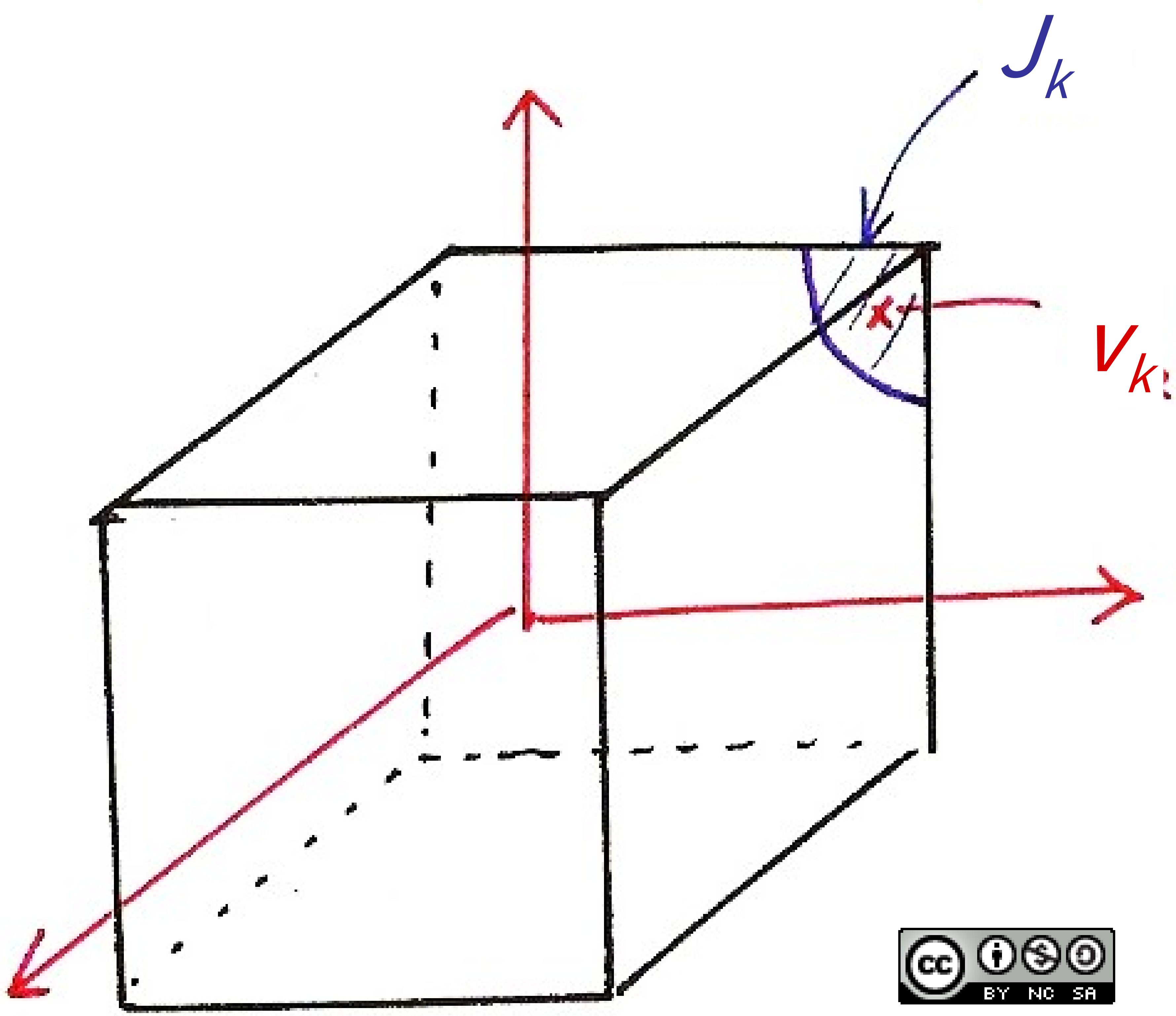
Vector Quantization



$$(256)^2=(2^8)^2=2^{16}$$

$$1024=2^{10}$$

Vector Quantization



Vector Quantization (VQ)

N-dim Vector Quantization

$\bar{x} = (x_1, x_2, \ldots, x_N)$

$S = \{\bar{x} = (x_1, x_2, \ldots, x_N), \; |x_k| \leq A, \; k = 1,2,\ldots N\}$

$S = \bigcup_{k=1}^{L} J_k$

$V = \{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_L\}$

$Q : S \rightarrow V$

$Q(\bar{x}) = \bar{v}_k$ if $\bar{x} \in J_k$

$L = 2^R$, each $\bar{v}_k$ represented by an R-bit pattern

Codebook Trained by a Large Training Set

- **Define distance measure between two vectors $\bar{x}$, $\bar{y}$**

$d(\bar{x}, \bar{y}) : S \times S \rightarrow R^+$ (non-negative real numbers)

-desired properties

$d(\bar{x}, \bar{y}) \geq 0$

$d(\bar{x}, \bar{x}) = 0$

$d(\bar{x}, \bar{y}) = d(\bar{y}, \bar{x})$

$d(\bar{x}, \bar{y}) + d(\bar{y}, \bar{z}) \geq d(\bar{x}, \bar{z})$

examples :

$d(\bar{x}, \bar{y}) = \sum_i (x_i - y_i)^2$

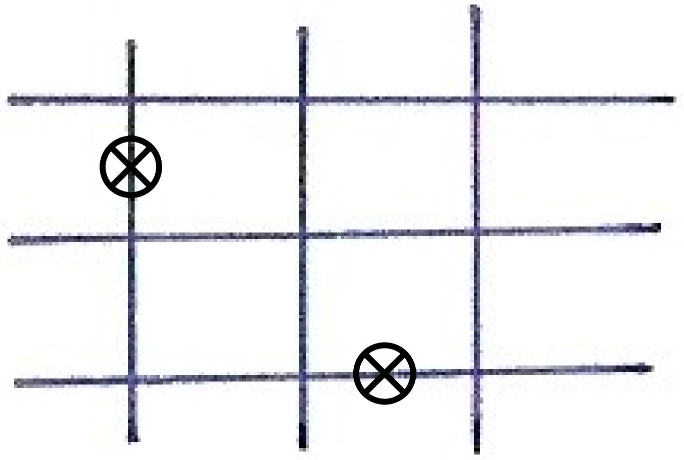
$d(\bar{x}, \bar{y}) = \sum_i | x_i - y_i |$

$d(\bar{x}, \bar{y}) = (\bar{x}-\bar{y})^t \Sigma^{-1} (\bar{x}-\bar{y})$

Distance Measures

$$\boxtimes(\overline{\boxtimes},\overline{\boxtimes})=\sum_{\boxtimes}\left|\boxtimes_{\boxtimes}-\boxtimes_{\boxtimes}\right|$$

city block distance



$$\boxtimes(\overline{\boxtimes},\overline{\boxtimes})=(\overline{\boxtimes}-\overline{\boxtimes})^{\boxtimes}\Sigma^{-1}(\overline{\boxtimes}-\overline{\boxtimes})$$

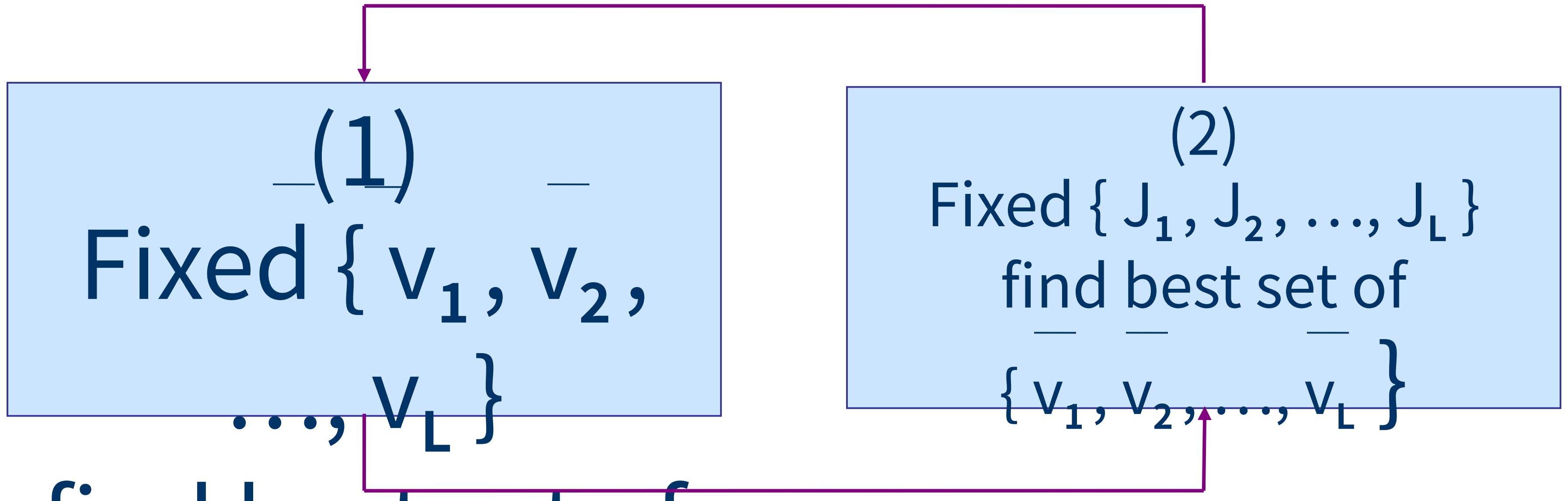
Mahalanobis distance

$$\Sigma\ \text{¿}\begin{bmatrix}1 & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & 1\end{bmatrix},\ \boxtimes(\overline{\boxtimes},\overline{\boxtimes})=\sum_{\boxtimes}(\boxtimes_{\boxtimes}-\boxtimes_{\boxtimes})^2$$

$$\Sigma\ \text{¿}\begin{bmatrix}\boxtimes_1^2 & \overset{\cdots}{\boxtimes_2^2} & 0 & \cdots\text{¿}\,\boxtimes_{\boxtimes}^2\text{¿}\\ \vdots & \text{¿} & \text{-¿} & \end{bmatrix},\ \boxtimes(\overline{\boxtimes},\overline{\boxtimes})=\sum_{\boxtimes}\frac{(\boxtimes_{\boxtimes}-\boxtimes_{\boxtimes})^2}{\boxtimes_{\boxtimes}^2}$$

Vector Quantization (VQ)

- **K-Means Algorithm/Lloyd-Max Algorithm**



(1) Fixed { $\bar{v}_1$, $\bar{v}_2$, …, $\bar{v}_L$ } find best set of { J_1 , J_2 , …, J_L }

(2) Fixed { J_1, J_2, …, J_L } find best set of { $\bar{v}_1$, $\bar{v}_2$, …, $\bar{v}_L$ }

(1) $J_k = \{ \bar{x} \mid d(\bar{x}, \bar{v}_k) < d(\bar{x}, \bar{v}_j), j \neq k \}$

$\rightarrow D = \sum_{all\ \bar{x}} d(\bar{x}, Q(\bar{x})) = \min$

nearest neighbor condition

(2) For each k

$\bar{v}_k = \frac{1}{M} \sum_{\bar{x} \in J_k} \bar{x}$

$\sum_{\bar{x} \in J_k}$

(3) Convergence condition

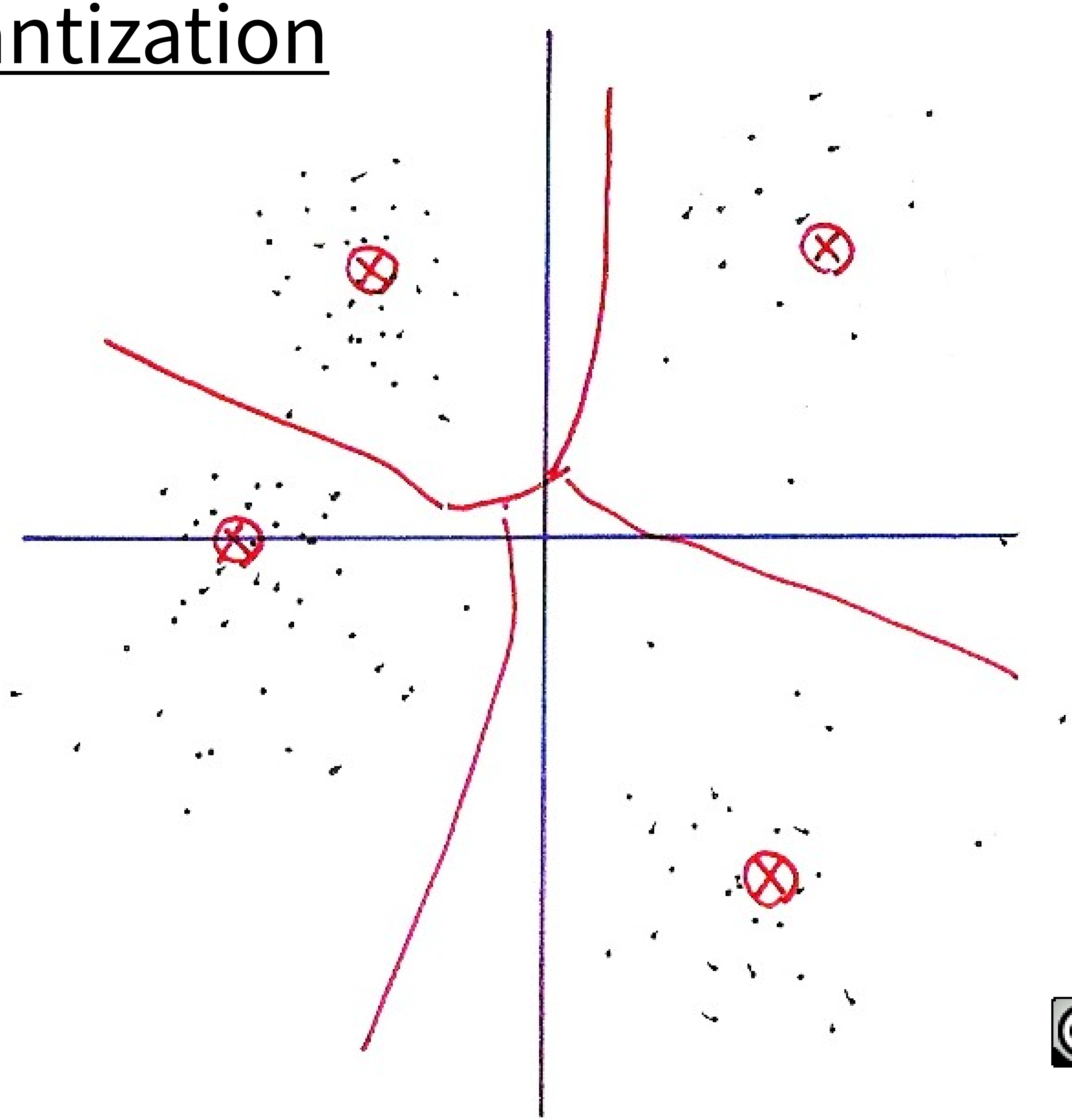
$D = \sum_{k=1}^{L} D_k$

after each iteration D is reduced, but $D \geq 0$

$| D^{(m+1)} - D^{(m)} | < \in$, m : iteration

- **Iterative Procedure to Obtain Codebook from a Large Training Set**

Vector Quantization



Vector Quantization (VQ)

- **K-means Algorithm may Converge to Local Optimal Solutions**
  - depending on initial conditions, not unique in general
- **Training VQ Codebook in Stages— LBG Algorithm**
  - step 1: Initialization. L = 1, train a 1-vector VQ codebook

$$\bar{v} = \frac{1}{N}\sum_j x'_j$$

  - step 2: Splitting.

Splitting the L codewords into 2L codewords, L = 2L

- example 1

$$\bar{v}_k^{(1)} = \bar{v}_k(1+\varepsilon)$$

$$\bar{v}_k^{(2)} = \bar{v}_k(1-\varepsilon)$$

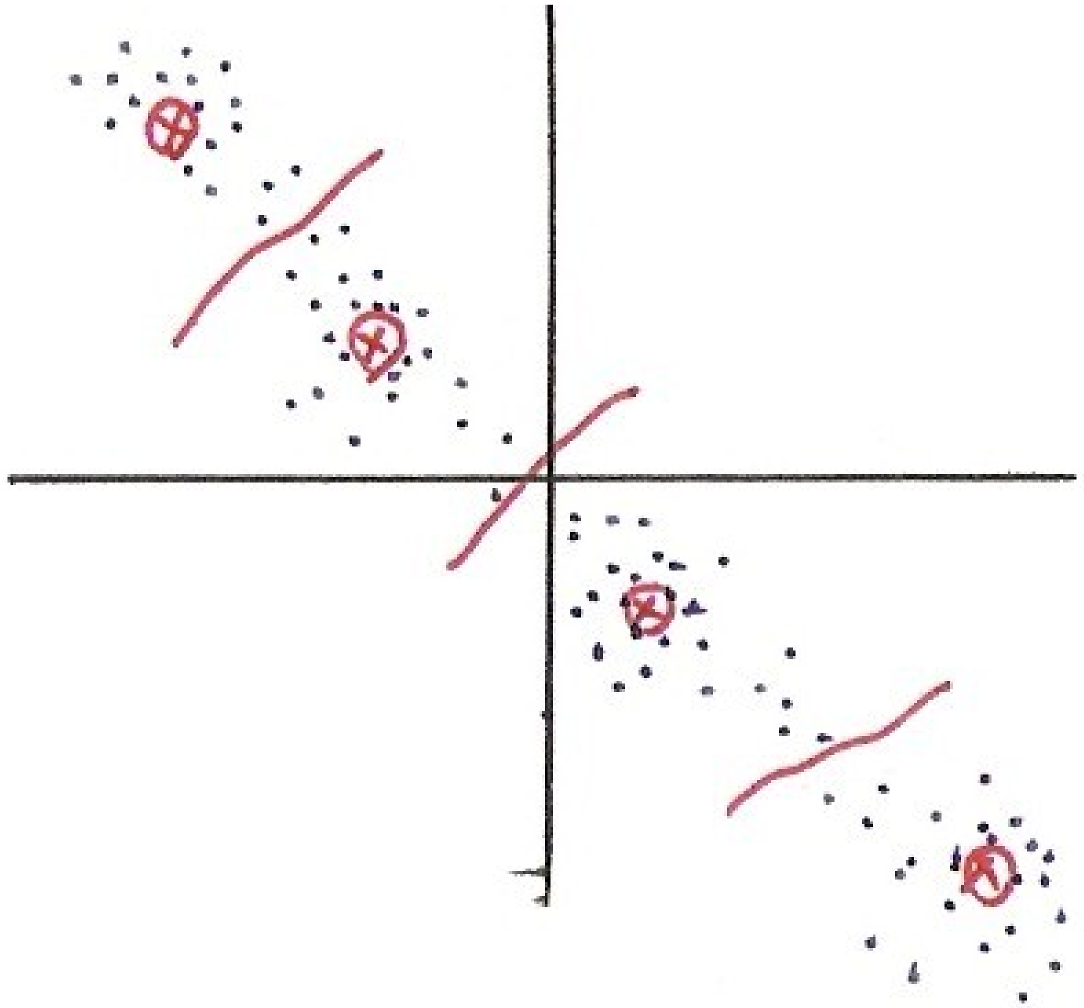
- example 2

$$\bar{v}_k^{(1)} = \bar{v}_k$$

$\bar{v}_k^{(2)}$: the vector most far apart

  - step 3: k-means Algorithm: to obtain L-vector codebook
  - step 4: Termination. Otherwise go to step 2
- **Usually Converges to Better Codebook**

LBG Algorithm



Initialization in HMM Training

- **An Often Used Approach— Segmental K-Means**
  - Assume an initial estimate of all model parameters (e.g. estimated by segmentation of training utterances into states with equal length)
    - For discrete density HMM

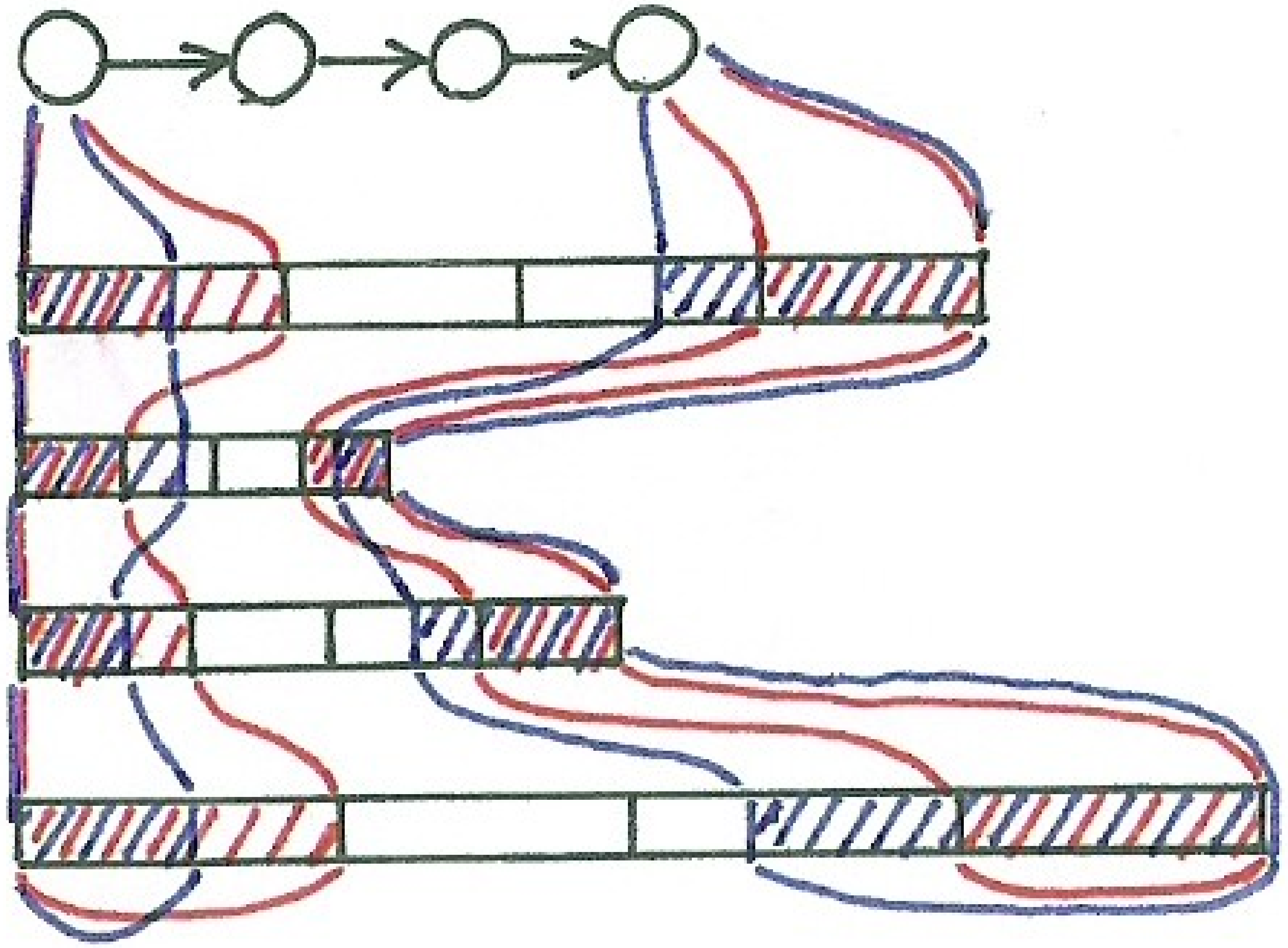
$$b_j(k)=\frac{\text{number of vectors in state } j \text{ associated with codeword } k}{\text{total number of vectors in state } j}$$

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

$$\begin{aligned}
&\Rightarrow \text{cluster the observation vectors within each state } j \text{ into a set of M clusters} \\
&\quad \text{(e.g. with vector quantization)} \\
&c_{jm} = \text{number of vectors classified in cluster } m \text{ of state } j \\
&\qquad \text{divided by number of vectors in state } j \\
&\mu_{jm} = \text{sample mean of the vectors classified in cluster } m \text{ of state } j \\
&\Sigma_{jm} = \text{sample covariance matrix of the vectors classified in cluster } m \text{ of state } j
\end{aligned}$$

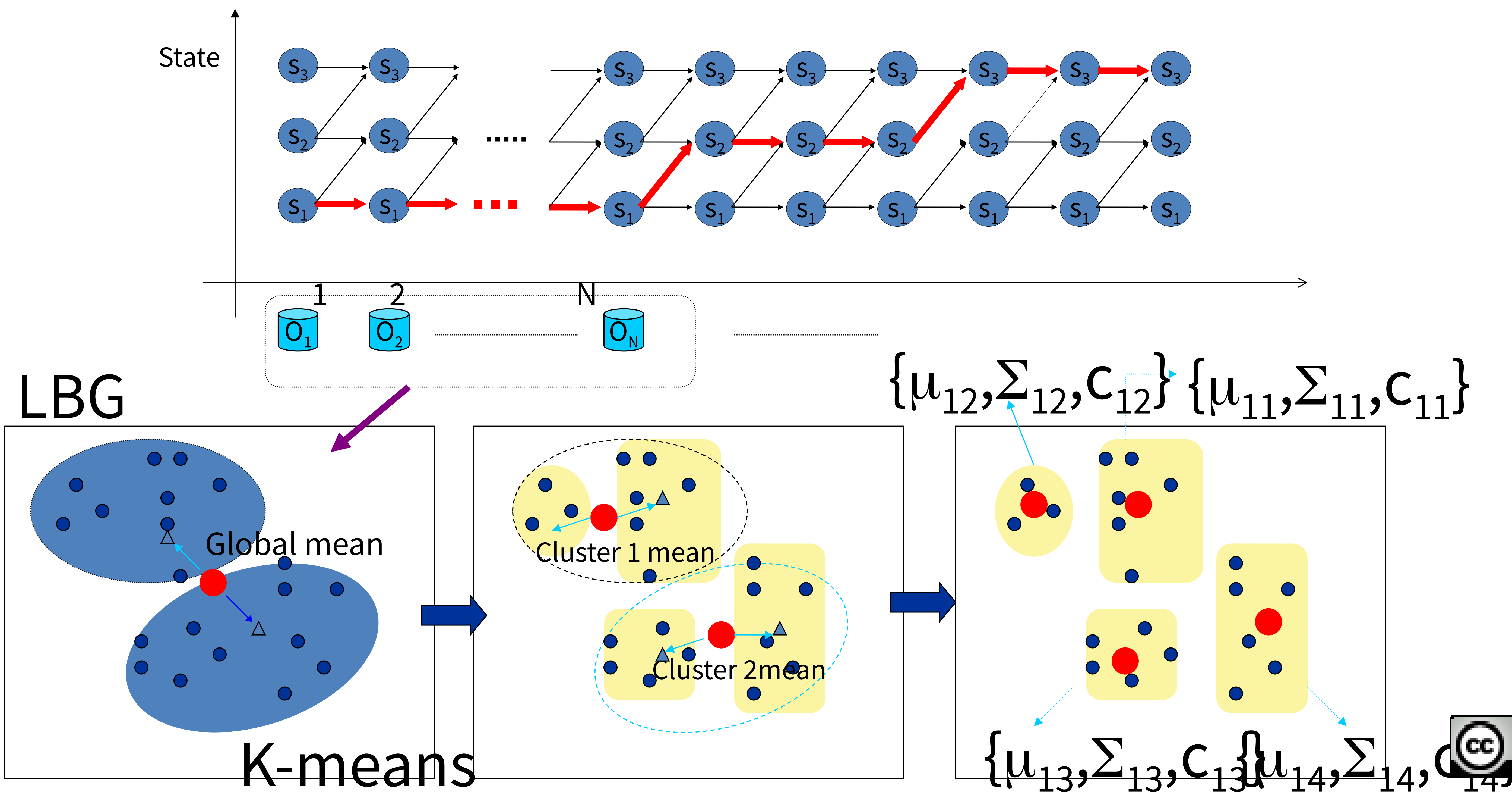
  - Step 1 : re-segment the training observation sequences into states based on the initial model by Viterbi Algorithm
  - Step 2 : Reestimate the model parameters (same as initial estimation)
  - Step 3: Evaluate the model score $P(\mathbf{O}|\lambda)$:
    If the difference between the previous and current model scores exceeds a threshold, go back to Step 1, otherwise stop and the initial model is obtained

Segmental K-Means



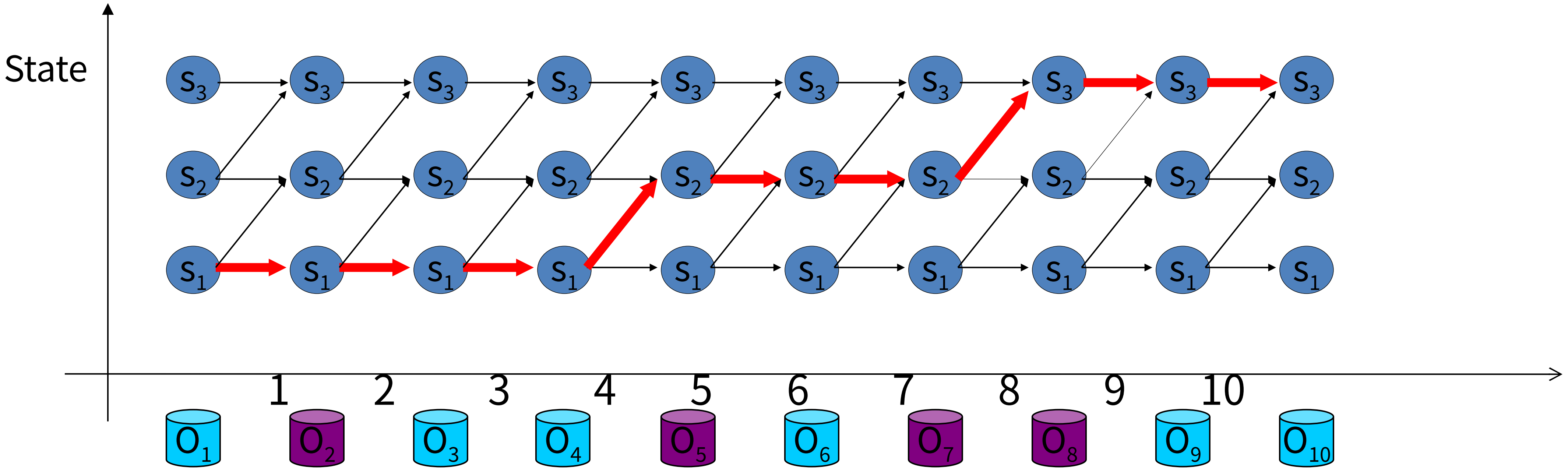
Initialization in HMM Training

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


State
s_3
s_2
s_1
.....
1
2
N
O_1
O_2
O_N
LBG
Global mean
Cluster 1 mean
Cluster 2mean
K-means
$\{\mu_{12},\Sigma_{12},c_{12}\}$
$\{\mu_{11},\Sigma_{11},c_{11}\}$
$\{\mu_{13},\Sigma_{13},c_{13}\}$
$\{\mu_{14},\Sigma_{14},c_{14}\}$

Initialization in HMM Training

- **An example for discrete HMM**
  - 3 states and 2 codewords


State
s_3 s_3 s_3 s_3 s_3 s_3 s_3 s_3 s_3 s_3
s_2 s_2 s_2 s_2 s_2 s_2 s_2 s_2 s_2 s_2
s_1 s_1 s_1 s_1 s_1 s_1 s_1 s_1 s_1 s_1
1 2 3 4 5 6 7 8 9 10
O_1 O_2 O_3 O_4 O_5 O_6 O_7 O_8 O_9 O_{10}

$\mathbf{v}_1$ 🟦

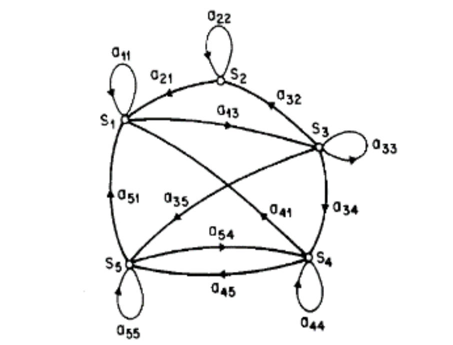
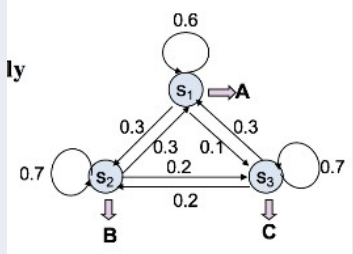
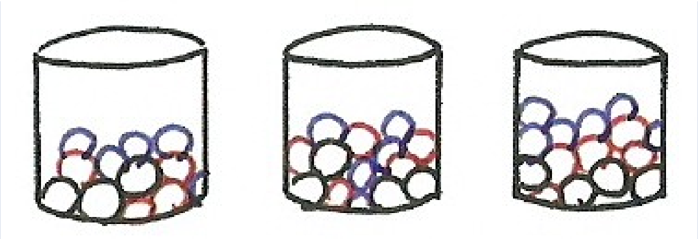
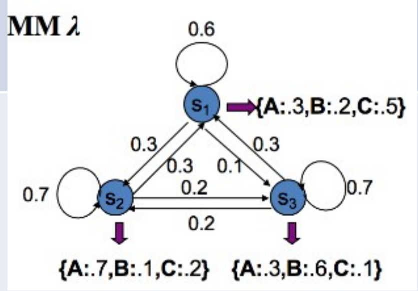
$\mathbf{v}_2$ 🟪

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

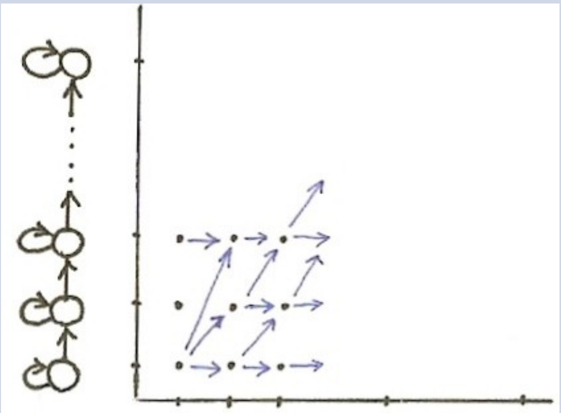
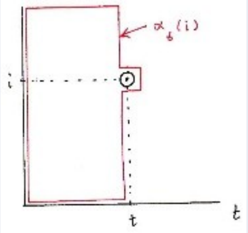
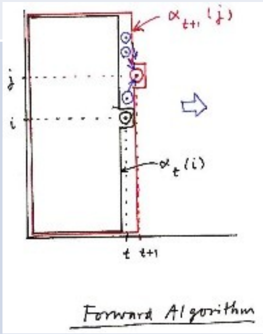
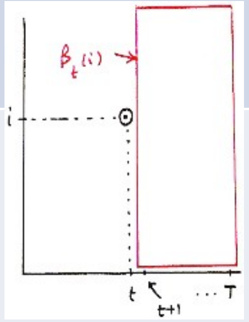
$b_2(\mathbf{v}_1)=1/3,\ b_2(\mathbf{v}_2)=2/3$

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

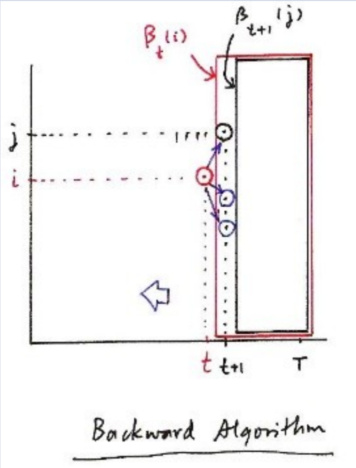
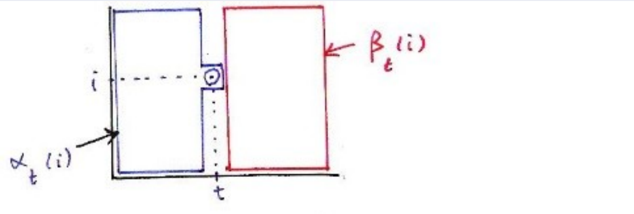
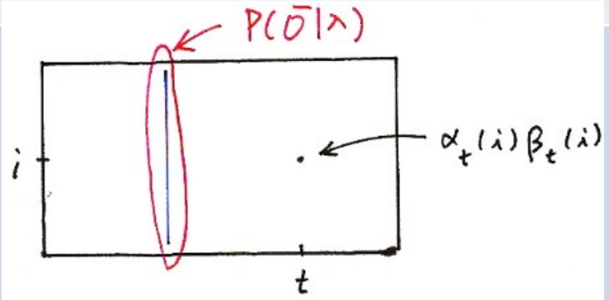
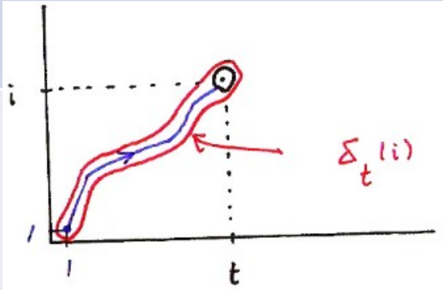
版權聲明

頁碼	作品	版權標示	作者 / 來源
2	 a_{22} a_{11} a_{21} S_2 a_{32} S_1 a_{13} S_3 a_{33} a_{51} a_{35} a_{41} a_{34} a_{54} S_5 S_4 a_{45} a_{55} a_{44} **Fig. 1.** A Markov chain with 5 states (labeled S_1 to S_5) with selected state transitions.		Lawrence Rabiner, Biing-Hwang Juang / FUNDAMENTALS OF SPEECH RECOGNITION Chap. 6, Sec. 6.2 Discrete-Time Markov Processes, page 323, Prentice-Hall International, Inc.
3	 0.6 ly S_1 A 0.3 0.3 0.3 0.1 0.7 0.2 0.7 S_2 S_3 0.2 B C	CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
5		CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
7	 MM λ 0.6 S_1 {A:.3,B:.2,C:.5} 0.3 0.3 0.3 0.1 0.7 0.2 0.7 S_2 S_3 0.2 {A:.7,B:.1,C:.2} {A:.3,B:.6,C:.1}	CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。

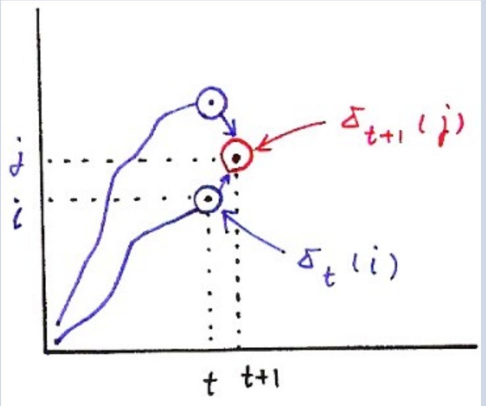
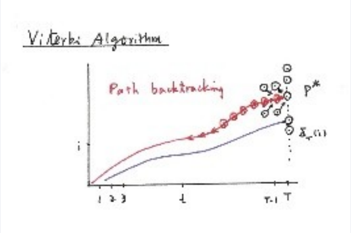
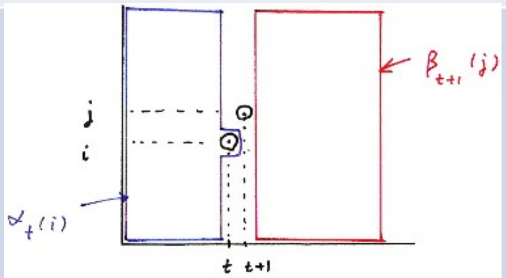
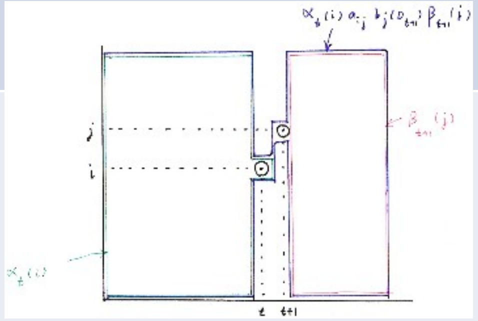
版權聲明

頁碼	作品	版權標示	作者 / 來源
11		CC BY NC SA	Lawrence Rabiner, Biing-Hwang Juang / FUNDAMENTALS OF SPEECH RECOGNITION Chap. 6, Sec. 6.2 Discrete-Time Markov Processes, page 323, Prentice-Hall International, Inc.
12		CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
13	 $\alpha_{t+1}(j)$ j i $\alpha_t(i)$ t t+1 Forward Algorithm	CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
15		CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。

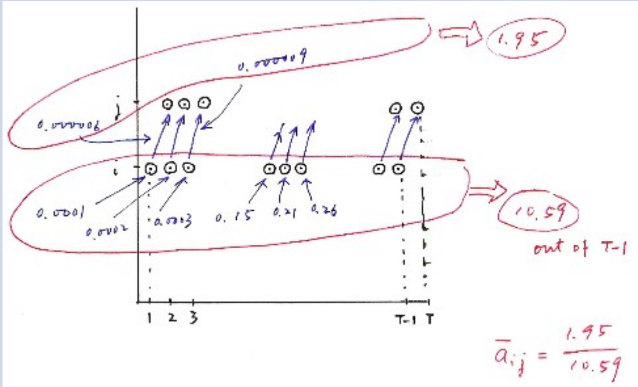
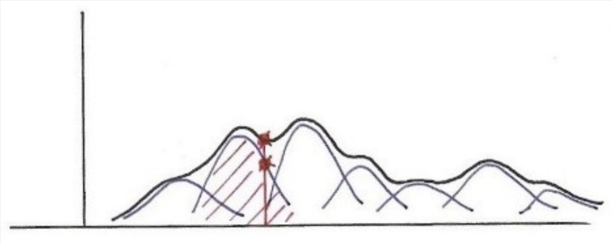
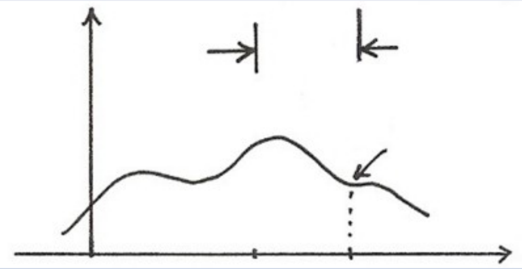
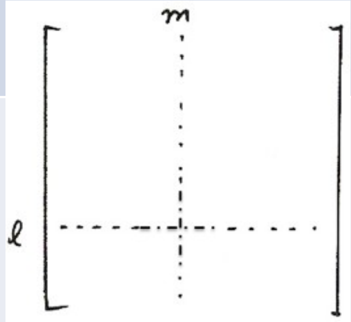
版權聲明

頁碼	作品	版權標示	作者 / 來源
16	 $\beta_t(i)$ $\beta_{t+1}(j)$ j i t t+1 T Backward Algorithm	cc BY NC SA	Lawrence Rabiner, Biing-Hwang Juang / FUNDAMENTALS OF SPEECH RECOGNITION Chap. 6, Sec. 6.2 Discrete-Time Markov Processes, page 323, Prentice-Hall International, Inc.
17		cc BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
18		cc BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
21		cc BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。

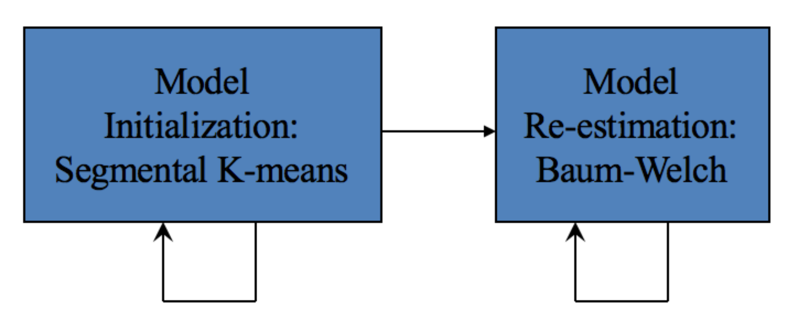
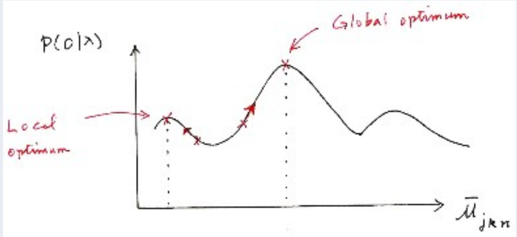
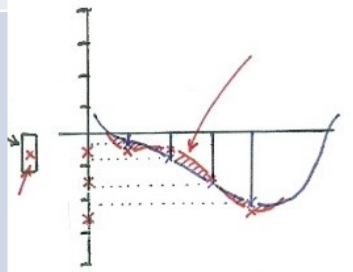
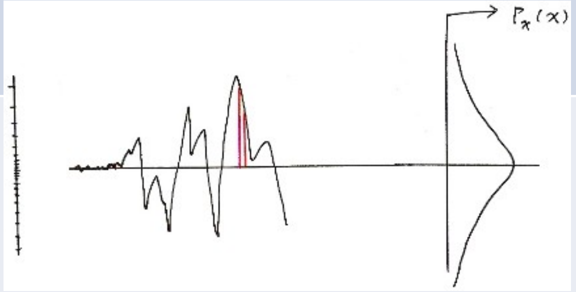
版權聲明

頁碼	作品	版權標示	作者 / 來源
21		cc BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
22	 Viterbi Algorithm Path backtracking P* i 1 2 3 t T	cc BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
26		cc BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
27	 $\alpha_t(i)\ a_{ij}\ b_j(o_{t+1})\ \beta_{t+1}(j)$ $\beta_{t+1}(j)$ j i $\alpha_t(i)$ t t+1	cc BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。

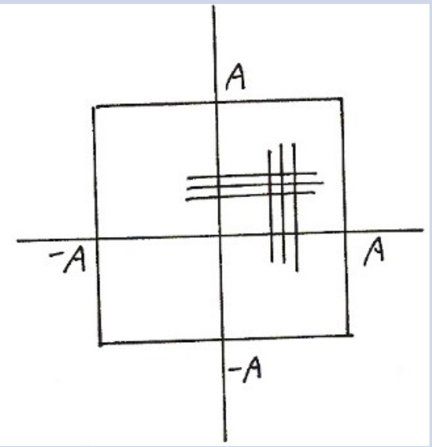
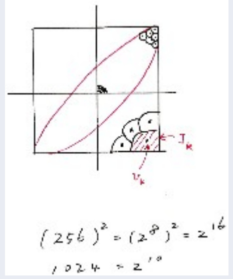
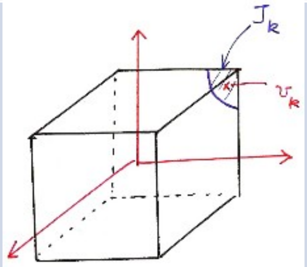
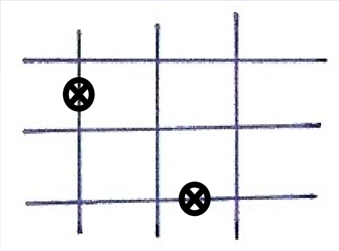
版權聲明

頁碼	作品	版權標示	作者 / 來源
29	 1.95 0.000019 0.000006 j i 0.0001 0.0002 0.0013 0.15 0.21 0.26 10.59 out of T-1 1 2 3 T-1 T $\bar{a}_{ij} = \frac{1.95}{10.59}$	CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
31		CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
33		CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
34		CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。

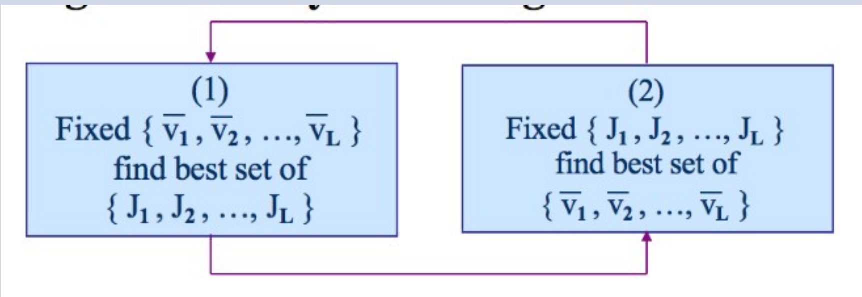
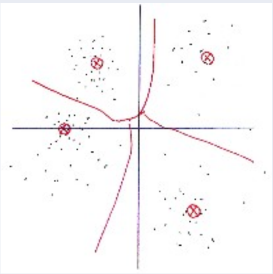
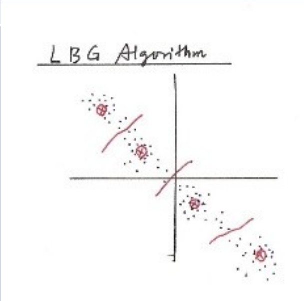
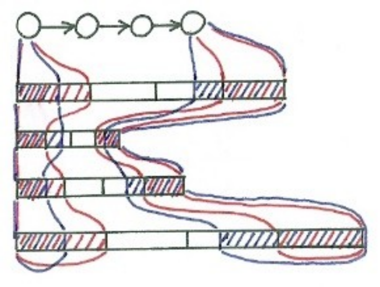
版權聲明

頁碼	作品	版權標示	作者 / 來源
35	 Model Initialization: Segmental K-means Model Re-estimation: Baum-Welch	CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
36		CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
38		CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
39		CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。

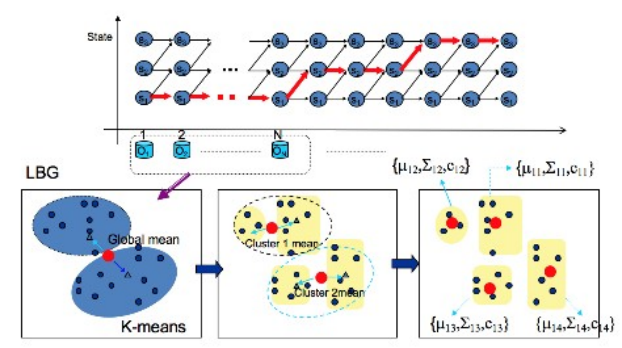
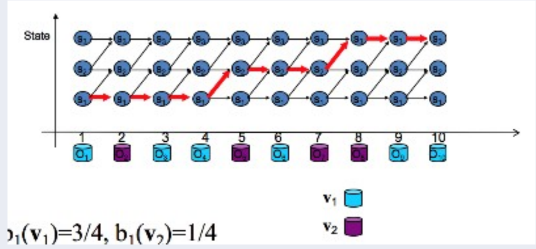
版權聲明

頁碼	作品	版權標示	作者 / 來源
41		CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
42	 J_k v_k $(256)^2 = (2^8)^2 = 2^{16}$ $1024 = 2^{10}$	CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
43		CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
45		CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。

版權聲明

頁碼	作品	版權標示	作者 / 來源
46	 (1) Fixed $\{ \bar{v}_1, \bar{v}_2, \ldots, \bar{v}_L \}$ find best set of $\{ J_1, J_2, \ldots, J_L \}$ (2) Fixed $\{ J_1, J_2, \ldots, J_L \}$ find best set of $\{ \bar{v}_1, \bar{v}_2, \ldots, \bar{v}_L \}$	cc BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
47		cc BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
49		cc BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
51		cc BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。

版權聲明

頁碼	作品	版權標示	作者 / 來源
52	 State LBG Global mean Cluster 1 mean Cluster 2 mean K-means $\{\mu_{12},\Sigma_{12},c_{12}\}$ $\{\mu_{11},\Sigma_{11},c_{11}\}$ $\{\mu_{13},\Sigma_{13},c_{13}\}$ $\{\mu_{14},\Sigma_{14},c_{14}\}$	CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。
53	 State 1 2 3 4 5 6 7 8 9 10 v_1 v_2 $b_1(v_1)$=3/4, $b_1(v_2)$=1/4	CC BY NC SA	國立臺灣大學電機工程學系李琳山 教授。 本作品採用創用 CC 「姓名標示 - 非商業性 - 相同方式分享 3.0 臺灣」許可協議。