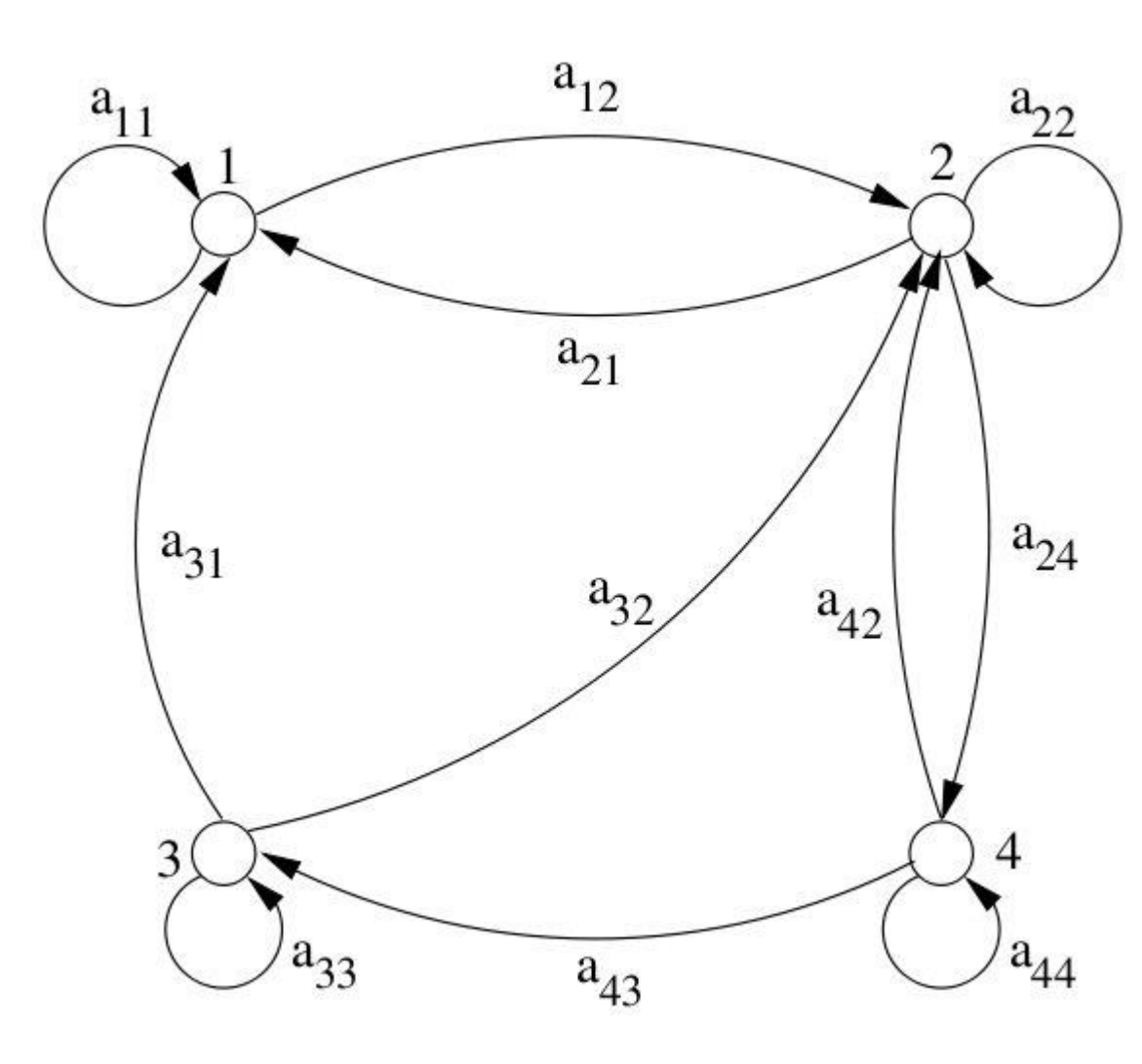
CONTENT-DEPENDENT CLASSIFICATION

MARKOV CHAINS

- Consider a system which at any time t is in one of a set of N distinct states $s_1, s_2, \ldots, s_N$.
- At regularly spaced discrete times, the system undergoes a change of state, according to a set of probabilities associated with the state.
- In Markovian processes, the probability of state transition depends only on the current and the previous state. Therefore we can adopt a diagram like in the following transparency for the transitions, where the a's represent the probabilities of transition between states:

$$a_{ij} = P(q_t = s_j \mid q_{t-1} = s_i)$$

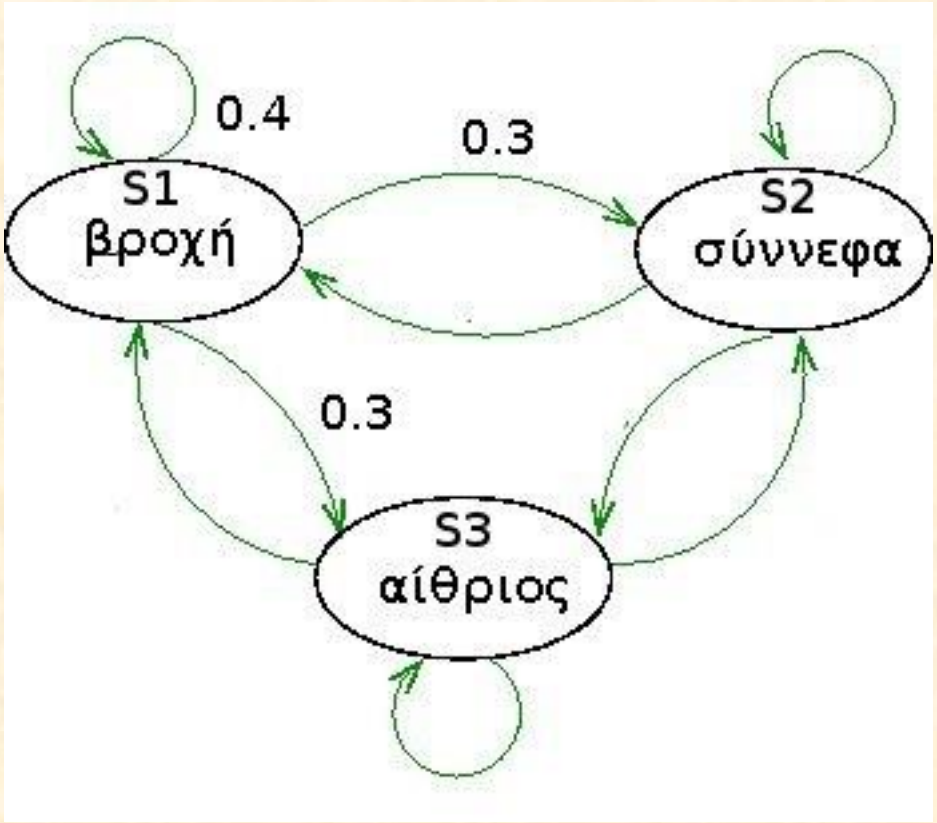
- Note: Each state corresponds to a physical (observable) event.


a11
a12
a22
1
2
a21
a31
a32
a42
a24
3
4
a33
a43
a44

Example: Weather sequences

- State 1: Rain or snow
- State 2: Cloudy weather
- State 3: Sunny weather

$$A = \{a_{ij}\} = \begin{bmatrix} 0,4 & 0,3 & 0,3 \\ 0,2 & 0,6 & 0,2 \\ 0,1 & 0,1 & 0,8 \end{bmatrix}$$



- Question: Given that on day 1 the weather is sunny, what is the probability of having the sequence:

Sun-sun-rain-rain-sun-cloud-sun in the next 7 days?

- Answer: Observation sequence: $O = S_3 S_3 S_3 S_1 S_1 S_3 S_2 S_3$

$$P(O \mid \text{Model}) =$$

$$P(S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3 \mid \text{Model}) =$$

$$P(S_3)P(S_3 \mid S_3)P(S_3 \mid S_3)P(S_1 \mid S_3)$$

$$P(S_1 \mid S_1)P(S_3 \mid S_1)P(S_2 \mid S_3)P(S_3 \mid S_2) =$$

Initial state probability

$$\pi_3 a_{33} a_{33} a_{31} a_{11} a_{13} a_{32} a_{23} =$$

$$1 * 0{,}8 * 0{,}8 * 0{,}1 * 0{,}4 * 0{,}3 * 0{,}1 * 0{,}2 = 1{,}536 * 10^{-4}$$

CONTEXT DEPENDENT CLASSIFICATION- OBSERVED MARKOV MODEL

- Remember: Bayes rule

$$P(\omega_i|\underline{x}) > P(\omega_j|\underline{x}),\ \forall j \neq i$$

- We will assume that the training vectors $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N$ occur in sequence, one after the other and we will refer to them as **observations**
- There is a strong dependence among classes (of a stochastic nature): Classes do not succeed each other at random, but there are specific probabilities for the succession of a class by another
- This interrelation demands the classification to be performed simultaneously for all available feature vectors and in the order that they appear as observations

❖ The Context Dependent Bayesian Classifier

- Let $X : \{\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N\}$
- Let $\omega_i, \; i = 1,2,\ldots,M$

N: Number of time steps
M: Number of classes

- Let Ω_i be a sequence of classes, that is

$$\Omega_i : \omega_{i1}\, \omega_{i2} \ldots \omega_{iN}$$

There are M^N of those

- Thus, the Bayesian rule can equivalently be stated as

$$X \to \Omega_i : \; P(\Omega_i | X) > P(\Omega_j | X) \;\; \forall i \neq j, \;\; i, j = 1,2,\ldots,M^N$$

❖ Markov Chain Models (for class dependence)

$$P(\omega_{i_k} | \omega_{i_{k-1}}, \omega_{i_{k-2}}, \ldots, \omega_{i_1}) = P(\omega_{i_k} | \omega_{i_{k-1}})$$

❖ NOW remember:

$$P(\Omega_i) = P(\omega_{i_1}, \omega_{i_2}, \ldots, \omega_{i_N}) =$$

$$= P(\omega_{i_N} \left| \omega_{i_{N-1}}, \ldots, \omega_{i_1} \right.).$$

$$P(\omega_{i_{N-1}} \left| \omega_{i_{N-2}}, \ldots, \omega_{i_1} \right.) \ldots P(\omega_{i_1})$$

or

$$P(\Omega_i) = \left(\prod_{k=2}^{N} P(\omega_{i_k} \left| \omega_{i_{k-1}} \right.)\right) P(\omega_{i_1})$$

❖ Assume:

- $\underline{x}_i$ statistically mutually independent
- The pdf in one class independent of the others, then

$$p(X \left| \Omega_i \right.) = \prod_{k=1}^{N} p(\underline{x}_k \left| \omega_{i_k} \right.)$$

- From the above, the Bayes rule is readily seen to be equivalent to:

$$P(\Omega_i|X)(><)P(\Omega_j|X)$$

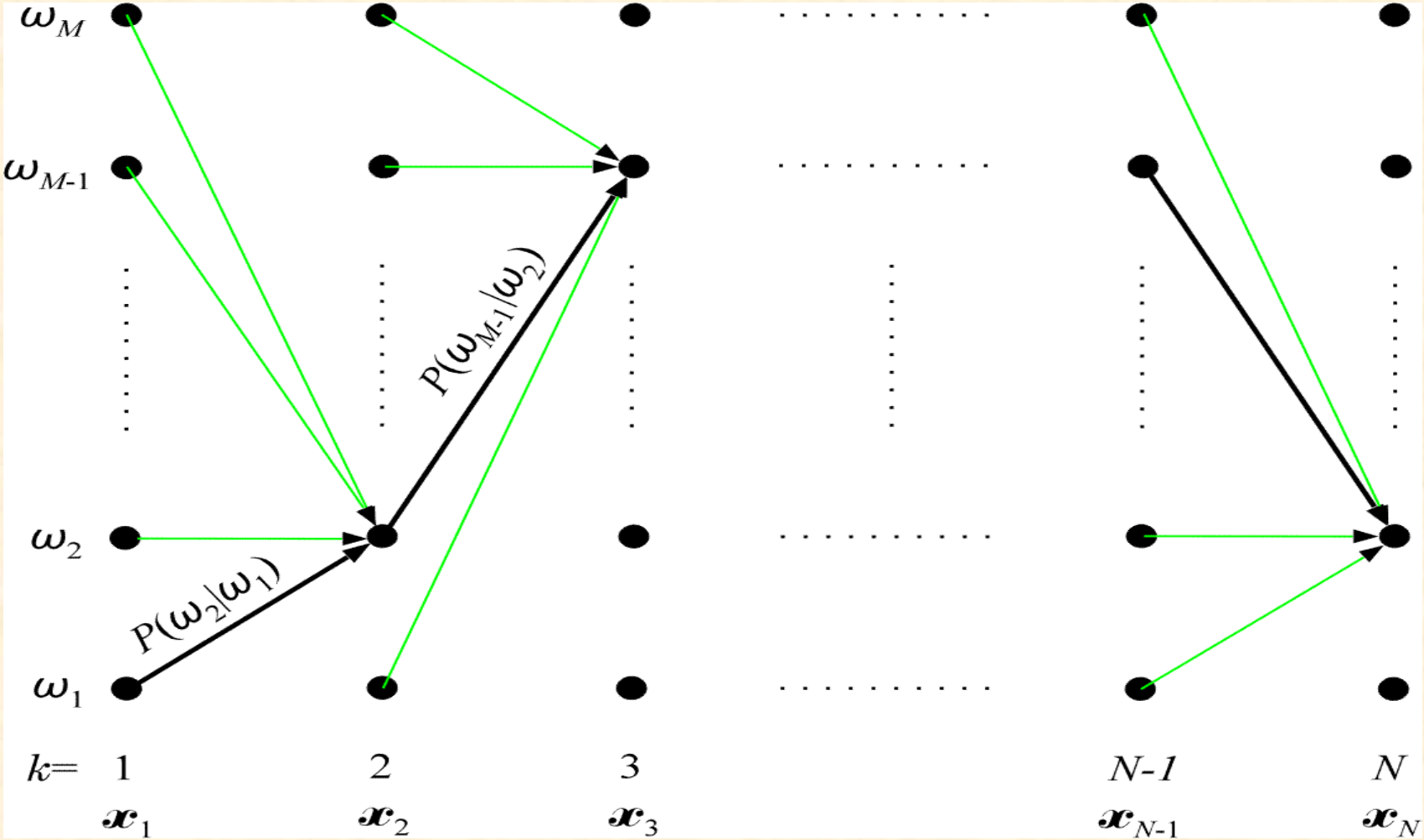
$$P(\Omega_i)p(X|\Omega_i)(><)P(\Omega_j)p(X|\Omega_j)$$

that is, it rests on

$$p(X|\Omega_i)P(\Omega_i) = P(\omega_{i_1})p(\underline{x}_1|\omega_{i_1}).$$
$$\prod_{k=2}^{N} P(\omega_{i_k}|\omega_{i_{k-1}})p(\underline{x}_k|\omega_{i_k})$$

- To find the above maximum in brute-force task we need $O(NM^N)$ operations!!

❖ The Viterbi Algorithm



- Thus, each Ω_i corresponds to one path through the trellis diagram. One of them is the optimum (e.g., black). The classes along the optimal path determine the classes to which ω_i are assigned.

$$p(X|\Omega_i)P(\Omega_i) = P(\omega_{i_1})\,p(\underline{x}_1|\omega_{i_1}).\prod_{k=2}^{N} P(\omega_{i_k}|\omega_{i_{k-1}})\,p(\underline{x}_k|\omega_{i_k})$$

- To each transition corresponds a cost. For our case

  - $\hat{d}(\omega_{i_k}, \omega_{i_{k-1}}) = P(\omega_{i_k}|\omega_{i_{k-1}}).p(\underline{x}_k|\omega_{i_k})$

  - $\hat{d}(\omega_{i_1}, \omega_{i_0}) \equiv P(\omega_{i_1})\,p(\underline{x}_i|\omega_{i_1})$

  - $\hat{D} = \prod_{k=1}^{N} \hat{d}(\omega_{i_k}, \omega_{i_{k-1}}) = p(X|\Omega_i)P(\Omega_i)$

- Equivalently

$$\ln \hat{D} = \sum_{k=1}^{N} \ln \hat{d}(.,.) \equiv D = \sum_{k=1}^{N} d(.,.)$$

where,

$$d(\omega_{i_k}, \omega_{i_{k-1}}) = \ln \hat{d}(\omega_{i_k}, \omega_{i_{k-1}})$$

- Define the cost up to a node ,*k*,

$$D(\omega_{i_k}) = \sum_{r=1}^{k} d(\omega_{i_r}, \omega_{i_{r-1}})$$

- Bellman's principle now states

$$D_{\max}(\omega_{i_k}) = \max_{i_{k-1}} \left[D_{\max}(\omega_{i_{k-1}}) + d(\omega_{i_k}, \omega_{i_{k-1}}) \right]$$

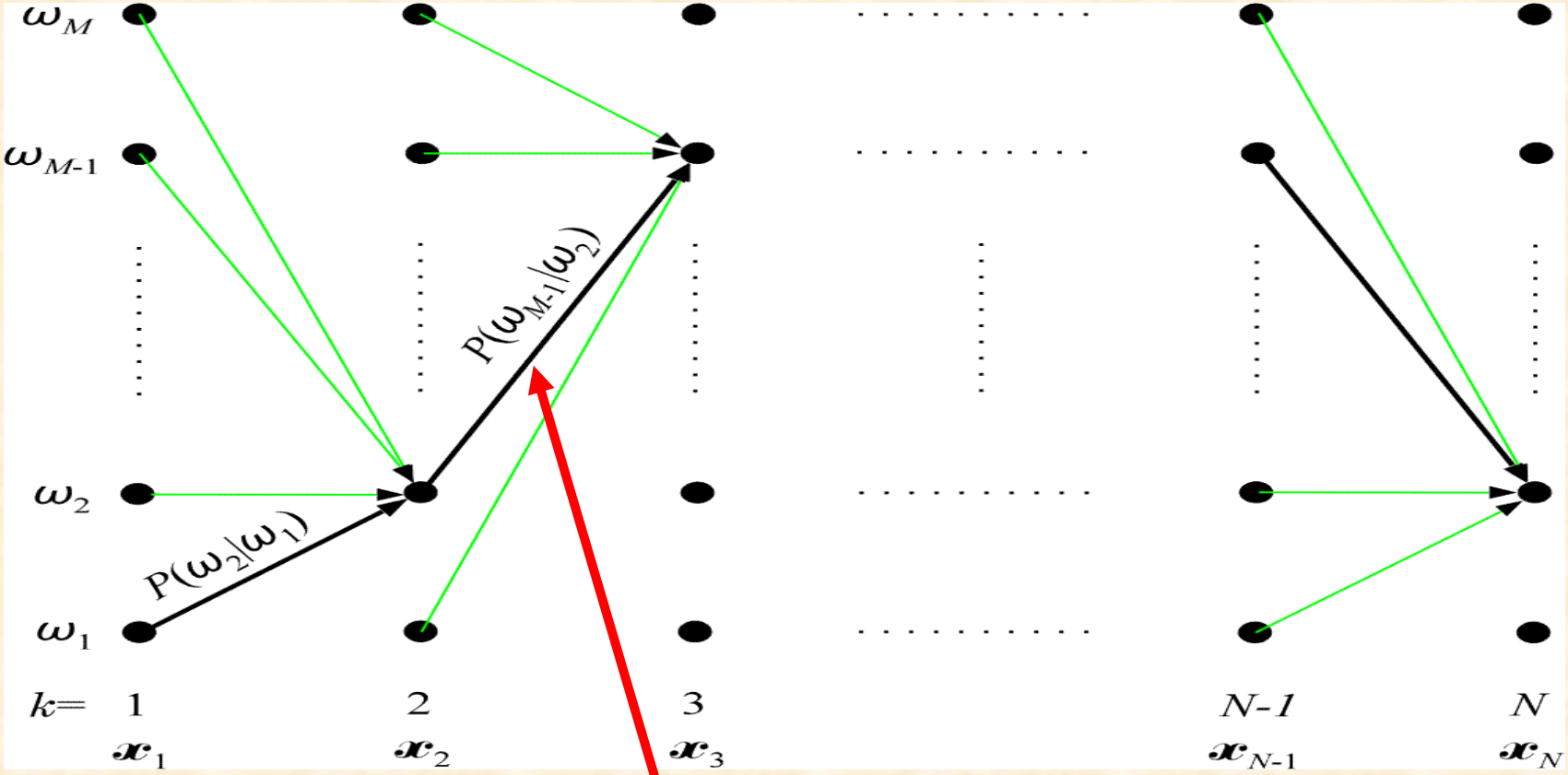
$$i_k, i_{k-1} = 1,2,\ldots,M$$

$$D_{\max}(\omega_{i_0}) = 0$$

- The optimal path terminates at ω^*_{iN} :

$$\omega^*_{i_N} = \arg\max_{\omega_{i_N}} D_{\max}(\omega_{i_N})$$

  - Complexity $O(NM^2)$



LOCAL COST:

$$\ln P(\omega_{M-1} \mid \omega_2) + \ln p(x_3 \mid \omega_{M-1})$$

❖ Channel Equalization

- The problem
- Recovering a transmitted sequence of information bits after they have been corrupted by the transmission channel and noise sources.

$$x_k = f(I_k, I_{k-1}, \ldots, I_{k-n+1}) + n_k$$

$$\underline{x}_k \equiv [x_k, x_{k-1}, \ldots, x_{k-l+1}]^T$$

$$\underline{x}_k \rightarrow \text{equalizer} \rightarrow \hat{I}_{k-r}$$

➢ Example

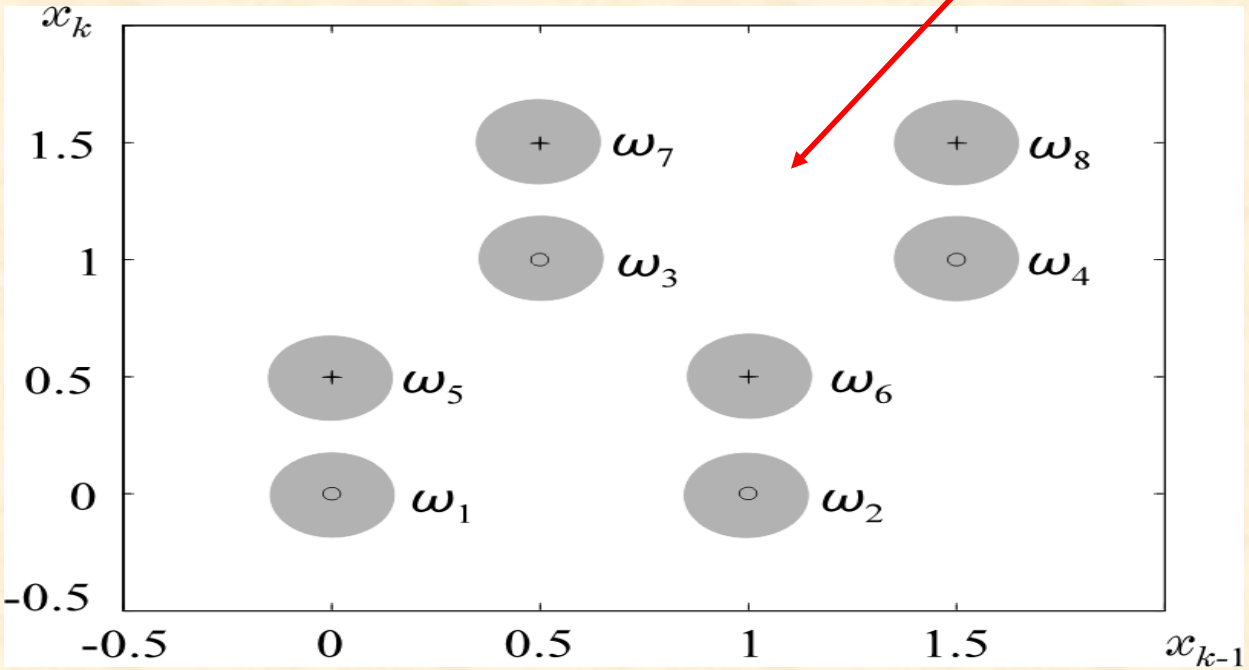
- $x_k = 0.5I_k + I_{k-1} + n_k$
- $\underline{x}_k = \begin{bmatrix} x_k \\ x_{k-1} \end{bmatrix}, l = 2$
- In $\underline{x}_k$ three input symbols are involved:

  I_k, I_{k-1}, I_{k-2}

without noise

I_k	I_{k-1}	I_{k-2}	x_k	x_{k-1}	
0	0	0	0	0	ω_1
0	0	1	0	1	ω_2
0	1	0	1	0.5	ω_3
0	1	1	1	1.5	ω_4
1	0	0	0.5	0	ω_5
1	0	1	0.5	1	ω_6
1	1	0	1.5	0.5	ω_7
1	1	1	1.5	1.5	ω_8

with noise



➢ Not all transitions are allowed

- $(I_k, I_{k-1}, I_{k-2}) = (0, 0, 1)$
- Then

$$(I_{k+1}, I_k, I_{k-1}) \begin{cases} (1, 0, 0) \\ (0, 0, 0) \end{cases}$$

- $\omega_2 \begin{cases} \omega_5 \\ \omega_1 \end{cases}$

$$P(\omega_2 | \omega_i) = \begin{cases} 0.5, i = 5, 1 \\ 0, \text{otherwise} \end{cases}$$

- In this context, ω_i are related to states. Given the current state and the transmitted bit, I_k, we determine the next state. The probabilities $P(\omega_i|\omega_j)$ define the state dependence model.

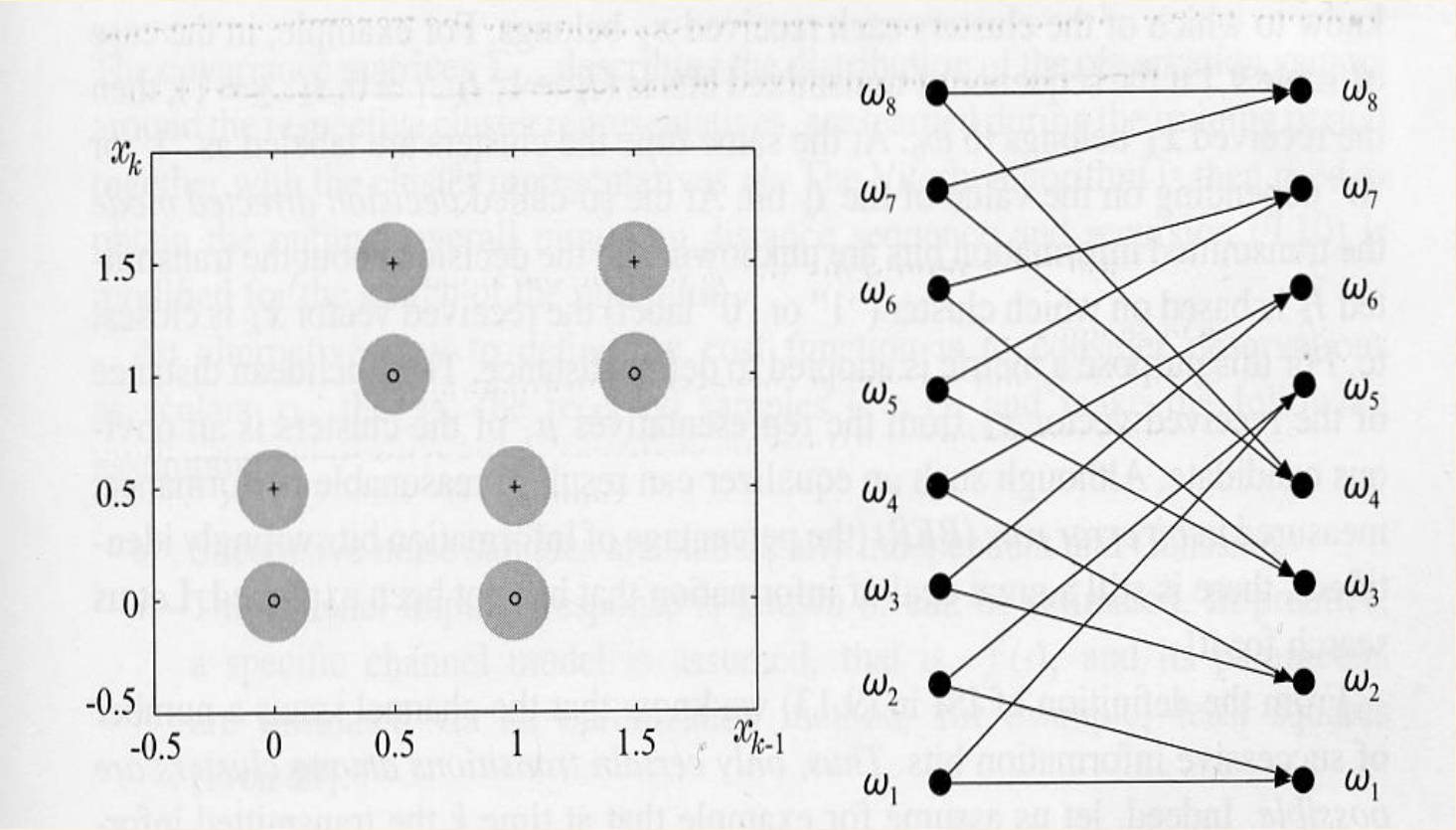
➢ Assuming Gaussian noise, the transition cost

- $$d(\omega_{i_k}, \omega_{i_{k-1}}) = d_{\omega_{i_k}}(\underline{x})$$

$$= \left\| \underline{x}_k - \underline{\mu}_{ik} \right\| = \left((\underline{x}_k - \underline{\mu}_{i_k})^T \sum\nolimits_{i_k}^{-1} (\underline{x}_k - \underline{\mu}_{i_k}) \right)^{\frac{1}{2}}$$

for all allowable transitions

Possible observations and allowable transitions between state

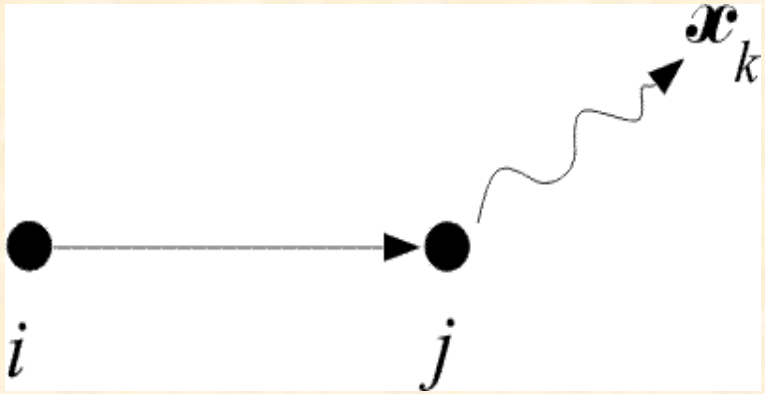


With this formulation, given a sequence of N observation vectors, we use the context dependent classifier to classify them in a sequence of clusters $\omega_{i_1}, \omega_{i_2}, \ldots, \omega_{i_N}$ This automatically classifies each vector, which is equivalent to deciding whether each I_k is 0 or 1.

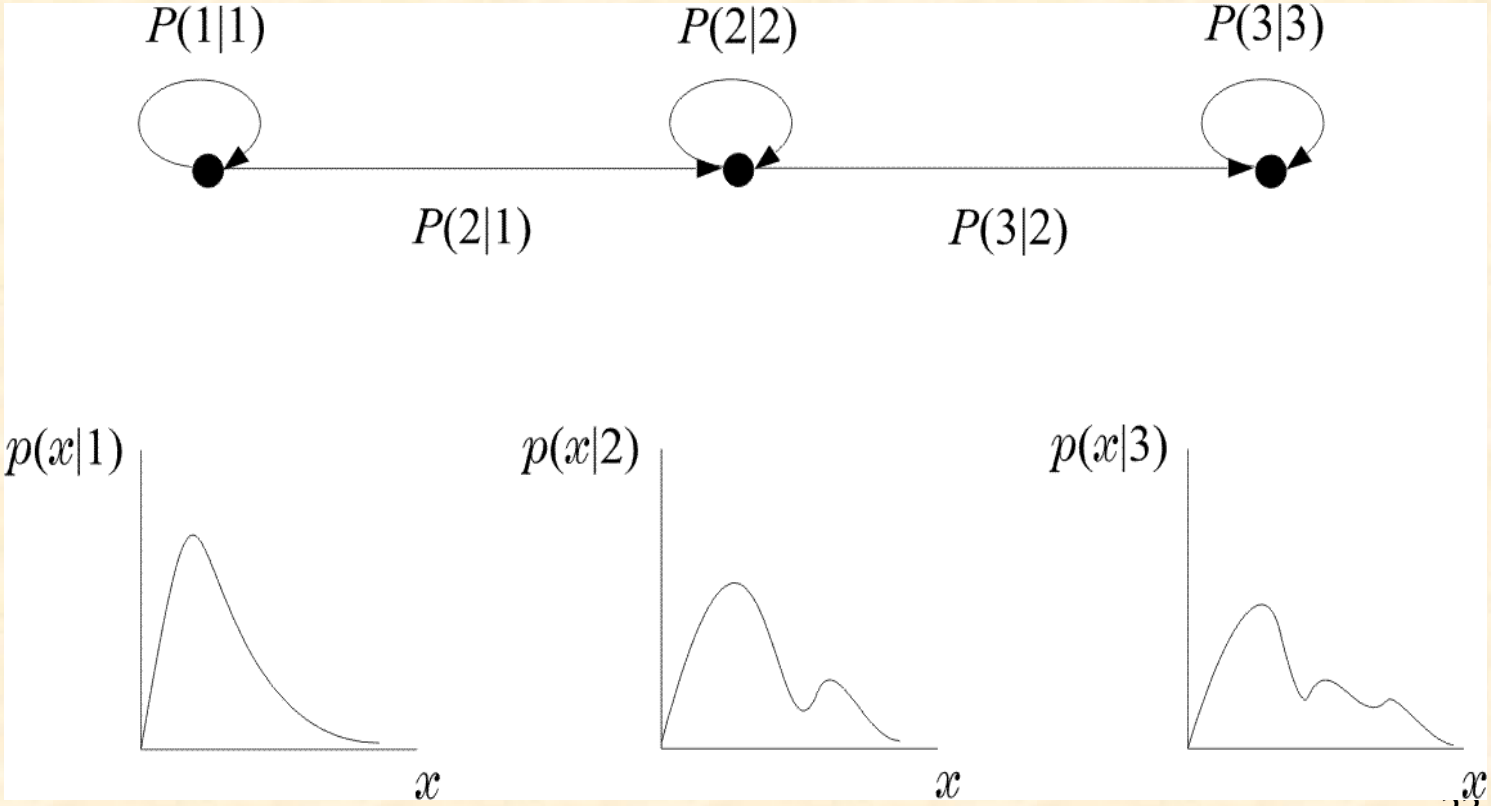
HIDDEN MARKOV MODELS

- In the channel equalization problem, the states are observable and can be "learned" during the training period
- Now we shall assume that states are not observable and can only be inferred from the training data
- Applications:
  - Speech and Music Recognition
  - OCR
  - Blind Equalization
  - Bioinformatics

- An HMM is a stochastic finite state automaton, that generates the observation sequence, $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N$
- We assume that: The observation sequence is produced as a result of **successive** transitions between states, upon arrival at a state:



- There is an underlying stochastic process which determines the transitions between states as before, but this process in not observable.
- It can only be inferred indirectly by the observation of another sets of stochastic processes that produce the sequence of observations.



- Examples of HMM:
  - The single coin case: Assume a coin that is tossed behind a curtain. All it is available to us is the outcome, i.e., H or T. Assume the two states to be:

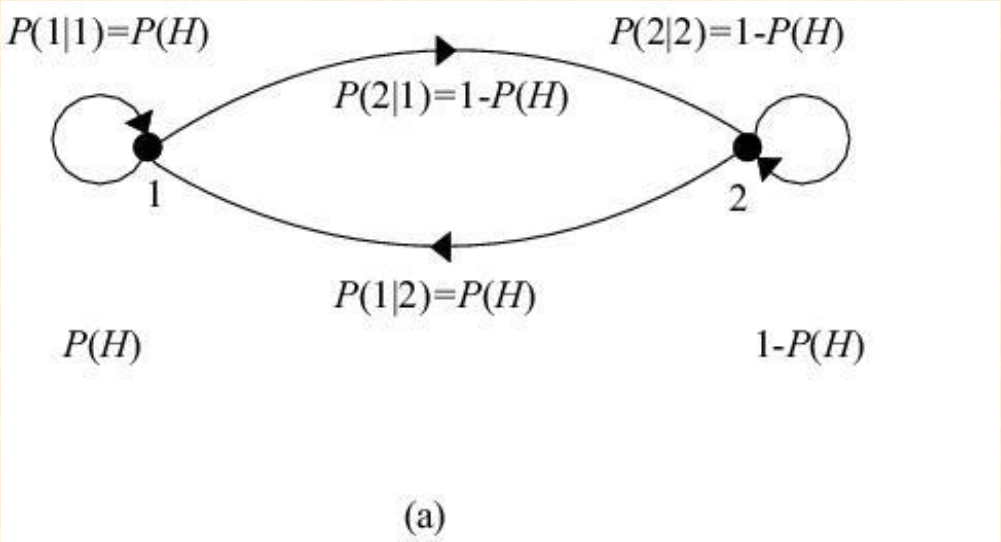
$$S = 1 \rightarrow H$$

$$S = 2 \rightarrow T$$

This is also an example of a random experiment with observable states. The model is characterized by a single parameter, e.g., $P(H)$. Note that

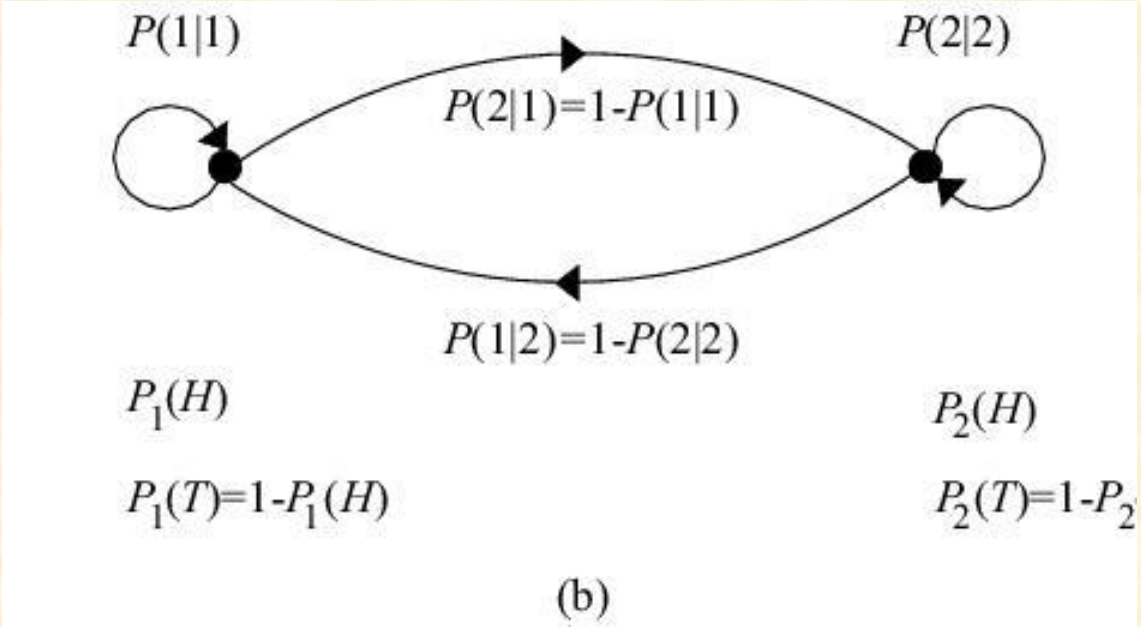
$$P(1|1) = P(H)$$

$$P(2|1) = P(\mathrm{T}) = 1 - P(H)$$

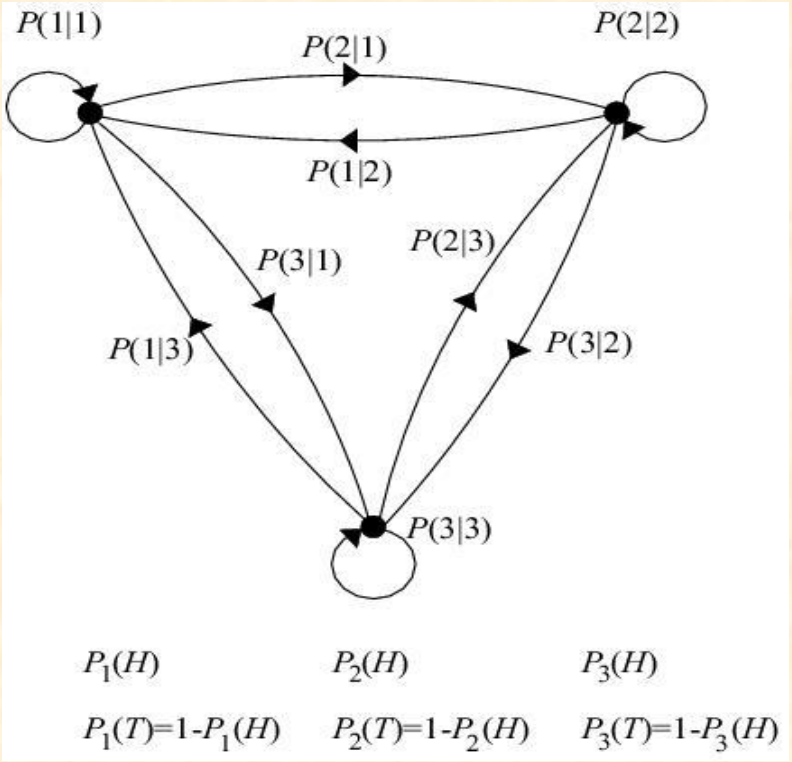


- The two-coins case: For this case, we observe a sequence of H or T. However, we have no access to know which coin was tossed. Identify one state for each coin. This is an example where states are not observable. H or T can be emitted from either state. The model depends on four parameters.

$$P_1(H),\ P_2(H),$$
$$P(1|1),\ P(2|2)$$

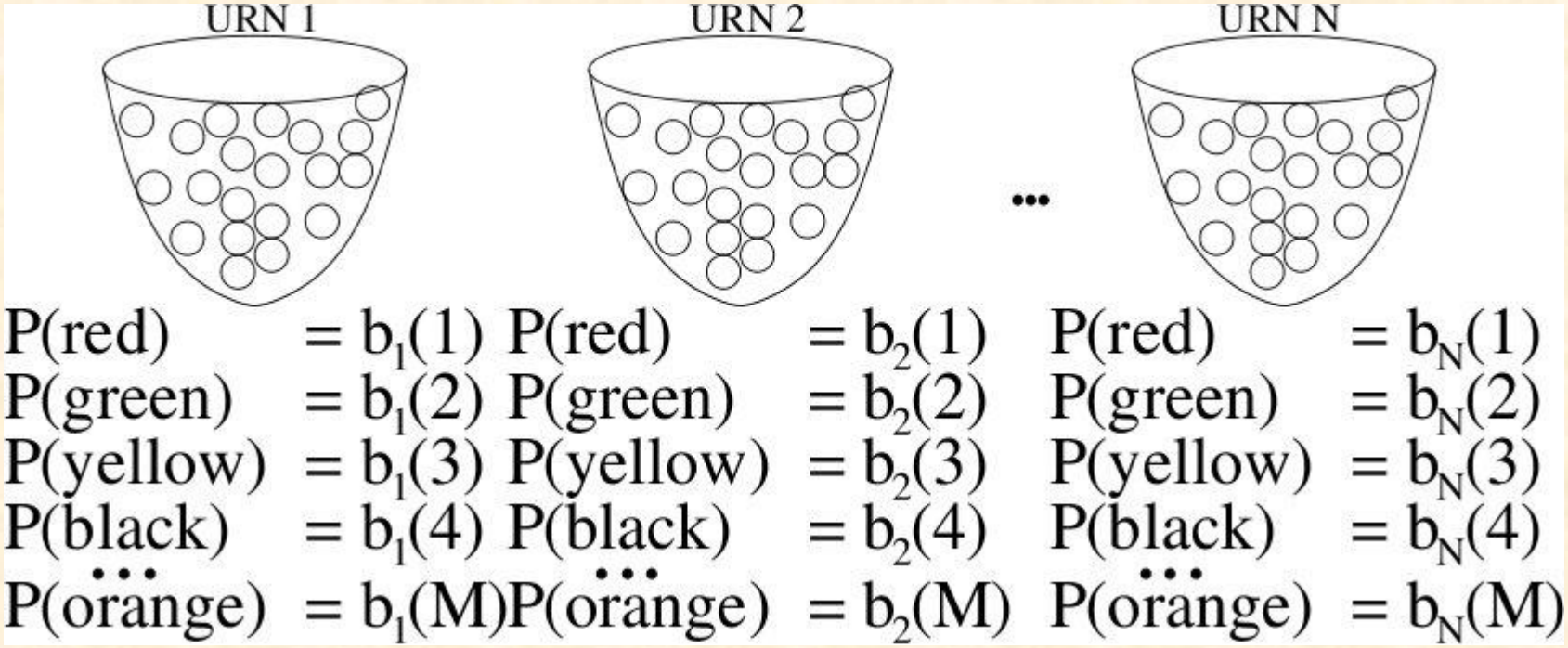


- The three-coins case example is shown below:



- Note that in all previous examples, specifying the model is equivalent to knowing:
  - The probability of each observation (H,T) to be emitted from each state.
  - The transition probabilities among states: $P(i|j)$.

URN AND BALL MODEL



- Balls of different colours are drawn out of one urn at a time
- The urn from which we draw at each time is decided by a Markovian process

- A general HMM model is characterized by the following set of parameters

  - K, number of states
  - M, number of observations symbols per state
  - $P(i|j), i, j = 1,2,\ldots,K \longrightarrow a_{ji}$
  - $p(\underline{x}|i), i = 1,2,\ldots,K \longrightarrow b_i(x)$
  - $P(i), i = 1,2,\ldots,K, \longrightarrow \pi_i$

    initial state probabilities

THE THREE BASIC PROBLEMS OF HMMs

- Evaluation: Given observation sequence $X = \underline{x}_1 \underline{x}_2 \underline{x}_3 \cdots \underline{x}_N$, and the HMM model S, compute $P(X \mid S)$, i.e. the probability of the observation sequence, given the model
- Discovery: Given observation sequence $X = \underline{x}_1 \underline{x}_2 \underline{x}_3 \cdots \underline{x}_N$, and the model S, choose a state sequence $Q = q_1 q_2 q_3 \cdots q_N$, which is optimal, i.e. it explains the observations in the best possible way
- Training: Given a set of observations, adjust the model parameters S to maximize $P(X \mid S)$

- Problem 1 is also related to recognition: given an observation sequence and L different HMMs (each corresponding to a category), choose the HMM for which is maximum $P(X \mid S)$.
- We will first see that problem 2 is also related to recognition, though in a suboptimal way.

Probability of a given observation sequence:
Similar to the observed Markov process, but now we must sum over all possible states:

$$P(X \mid S) = \sum_{\text{all Q}} P(X \mid Q, S) P(Q, S)$$

$$P(X \mid Q, S) = \prod_{k=1}^{N} P(\underline{x}_k \mid q_k, S)$$

$$P(X \mid Q, S) = b_{q_1}(\underline{x}_1) b_{q_2}(\underline{x}_2) \ldots b_{q_N}(\underline{x}_N)$$

$$P(Q, S) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \ldots a_{q_{N-1} q_N}$$

$$P(X \mid S) = \sum_{\text{all Q}} \pi_{q_1} b_{q_1}(\underline{x}_1) a_{q_1 q_2} b_{q_2}(\underline{x}_2) a_{q_2 q_3} \ldots a_{q_{N-1} q_N} b_{q_N}(\underline{x}_N)$$

RECOGNITION: BEST PATH METHOD (PROBLEM 2)

- Suboptimal approach.
- For a given observation sequence, we compute the most probable (best) path of states sequence, for each of the reference HMMs.
- The search of the optima is performed using directly the Viterbi algorithm with cost:

local cost

$$D = \sum_{k=1}^{N} d(q_k, q_{k-1})$$

$$d(q_k, q_{k-1}) = \ln P(q_k \mid q_{k-1}) + \ln\, p(x_k \mid q_k)$$

$$= \ln a_{q_{k-1}q_k} + \ln b_{q_k}(x_k)$$

RECOGNITION: ANY PATH METHOD (PROBLEM 1)

- Assume the L models to be known (L classes).
- A sequence of observations, X, is given.
- Assume observations to be emissions upon the arrival on successive states
- Decide in favor of the model S^* (from the L available) according to the Bayes rule

$$S^* = \arg\max_S P(S|X)$$

for equiprobable patterns

$$S^* = \arg\max_S p(X|S)$$

- For the efficient computation of $P(X \mid S)$ define:

$$A(q_k) = P(\underline{x}_1 \underline{x}_2 \ldots \underline{x}_k, q_k \mid S) \longrightarrow P(X \mid S) = \sum_{q_N=1}^{K} A(q_N)$$

- Note that then:

$$P(X \mid S) = \sum_{q_N=1}^{K} A(q_N)$$

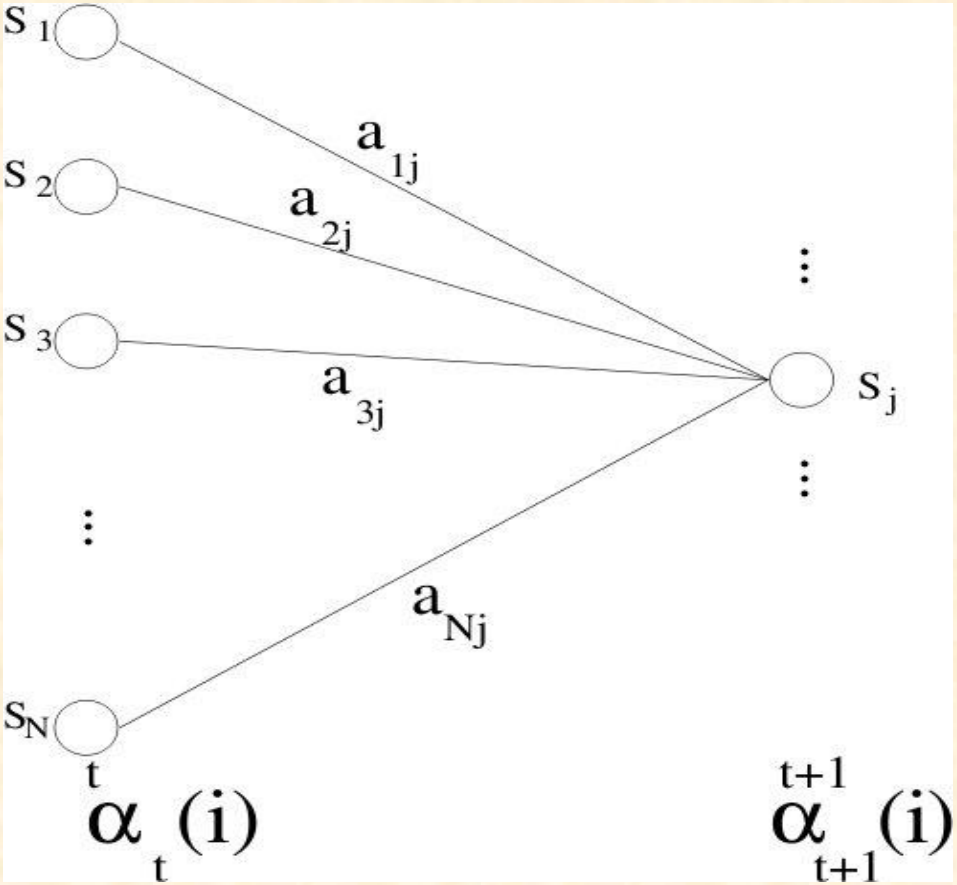
- To get A at time slice k, we must sum over possible states occuring at time slice k-1:

$$A(q_k) = \sum_{i_{k-1}} A(q_{k-1}) P(q_k \mid q_{k-1}) P(\underline{x}_k \mid q_k) =$$

History

$$= \sum_{i_{k-1}} A(q_{k-1}) a_{q_{k-1} q_k} b_{q_k}(\underline{x}_k)$$

Local activity



Therefore, each A at the next time slice is computed from all As in the previous time slice by a constructive procedure. K^2N calculations are enough to achieve the whole computation.

- Some more quantities
  - $$B(q_k) = p(\underline{x}_{k+1}, \underline{x}_{k+2}, \ldots, \underline{x}_N \mid q_k, S)$$
    $$= \sum_{q_{k+1}} B(q_{k+1}) P(q_{k+1} \mid q_k)\, p(\underline{x}_{k+1} \mid q_{k+1})$$
  - $$\Gamma(q_k) = p(\underline{x}_1, \ldots, \underline{x}_N, q_k \mid S)$$
    $$= \mathrm{A}(q_k)\mathrm{B}(q_k)$$

- Training
  - The philosophy:

    Given a training set X, known to belong to the specific model, estimate the unknown parameters of S, so that the **output** of the model, e.g.

$$P(X \mid S) = \sum_{q_N=1}^{K} A(q_N)$$

    to be maximized

- This is a ML estimation problem with missing data

- Assumption: Data $\underline{x}$ discrete

$$\underline{x} \in \{1,2,\ldots,r\} \Rightarrow p(\underline{x}|i) \equiv P(\underline{x}|i)$$

- Definitions:

  - $$\xi_k(i,j) = \frac{A(i_k = i)P(j|i)P(\underline{x}_{k+1}|j)B(i_{k+1} = j)}{P(X|S)}$$

  - $$\Gamma_k(i) = \frac{A(i_k = i)B(i_k = i)}{P(X|S)}$$

➢ The Algorithm:

- Initial conditions for all the unknown parameters.
  Compute $P(X|S)$

- Step 1: From the current estimates of the model parameters reestimate the new model S from

  – $\overline{P}(j|i) = \dfrac{\sum_{k=1}^{N-1} \xi_k(i,j)}{\sum_{k=1}^{N-1} \Gamma_k(i)}$ $\left(= \dfrac{\#\text{ of transitions from } i \text{ to } j}{\#\text{ of transitions from } i} \right)$

  – $\overline{P}_{\underline{x}}(r|i) = \dfrac{\sum_{k=1 \text{ and } \underline{x} \to r)}^{N} \Gamma_k(i)}{\sum_{k=1}^{N} \Gamma_k(i)}$ $\left(= \dfrac{\text{at state } i \text{ and } \underline{x} = r}{\neq \text{ of being at state } i} \right)$

  – $\overline{P}(i) = \Gamma_1(i)$

- Step 3: Compute $P(X|\overline{S})$. If $P(X|\overline{S}) - P(X|S) > \varepsilon,\ S = \overline{S}$ go to step 2. Otherwise stop

- Remarks:
  - Each iteration improves the model

    $$\overline{S}:\ P(X|\overline{S}) > P(X|S)$$

  - The algorithm converges to a maximum (local or global)
  - The algorithm is an implementation of the EM algorithm