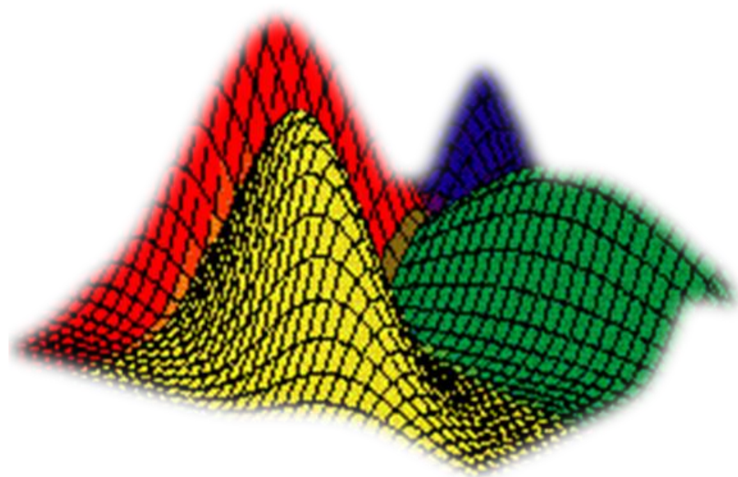


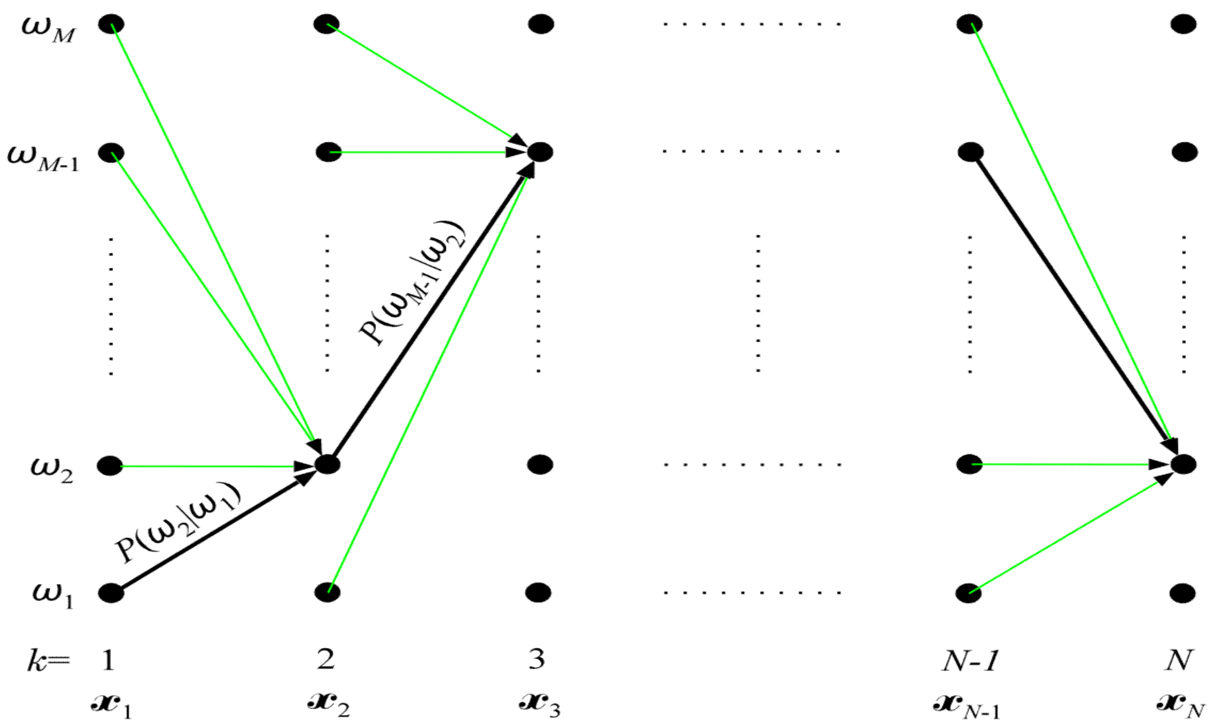
CSE 473
Pattern Recognition

Context Dependent Classification

Context Dependent Classification

- In Context dependent classification, the class of a feature vector depends on
  - Its own value
  - Value of other feature vectors
  - Classes assigned to other vectors

Viterbi Algorithm: Calculate $p(X|\Omega_i)p(\Omega_i)$



- Total M nodes in each of N columns
- Each node has M transitions
- Thus, complexity is $O(NM^2)$

Application in Channel Equalization

- The problem
  - Information bits I_k are transmitted
  - We receive x_k

$$I_k \rightarrow \boxed{\text{Channel}} \rightarrow x_k$$

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

where, f is the action of channel

n_k is the noise

Application in Channel Equalization

$$I_k \rightarrow \boxed{\text{Channel}} \rightarrow x_k$$

– We want the bits back

$$\underline{x}_k \rightarrow \boxed{\text{equalizer}} \rightarrow \hat{I}_k$$

where,

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

Application in Channel Equalization

$$I_k \rightarrow \boxed{\text{Channel}} \rightarrow x_k$$

– We want the bits back

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

OR $\underline{x}_k \rightarrow \hat{I}_k \text{ or } \hat{I}_{k-r}$ if we allow delay r

Application in Channel Equalization

$$I_k \rightarrow \boxed{\text{Channel}} \rightarrow x_k$$

– We want the bits back

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

OR $\underline{x}_k \rightarrow \hat{I}_k \text{ or } \hat{I}_{k-r}$ if we allow delay r

This means we use the vector $\underline{x}_k$ to get a single bit $\hat{I}_k \text{ or } \hat{I}_{k-r}$

Application in Channel Equalization

- 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 vector $\underline{x}_k$ three input symbols are involved:

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

Application in Channel Equalization

- Assuming n_k =0

I_k	I_{k-1}	I_{k-2}			$\hat{I}_k$
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Application in Channel Equalization

- Assuming n_k =0

		$\boldsymbol{I_{k-2}}$			$\hat{I}_k$
		0			
		1			
		0			
		1			
		0			
		1			
		0			
		1			

Application in Channel Equalization

- Assuming n_k =0

	I_{k-1}	I_{k-2}			$\hat{I}_k$
	0	0			
	0	1			
	1	0			
	1	1			
	0	0			
	0	1			
	1	0			
	1	1			

Application in Channel Equalization

- Assuming n_k =0

I_k	I_{k-1}	I_{k-2}			$\hat{I}_k$
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Application in Channel Equalization

- Assuming n_k =0

	I_{k-1}	I_{k-2}		x_{k-1}	
	0	0		0	
	0	1		1	
	1	0		0.5	
	1	1		1.5	
	0	0		0	
	0	1		1	
	1	0		0.5	
	1	1		1.5	

$$I_{k-1} \rightarrow \boxed{\text{Channel}} \rightarrow x_{k-1}$$

$$x_{k-1} = 0.5 I_{k-1} + I_{k-2}$$

Application in Channel Equalization

- Assuming n_k =0

I_k	I_{k-1}		x_k		
0	0		0		
0	0		0		
0	1		1		
0	1		1		
1	0		0.5		
1	0		0.5		
1	1		1.5		
1	1		1.5		

$$I_k \rightarrow \boxed{\text{Channel}} \rightarrow x_k$$

$$x_k = 0.5 I_k + I_{k-1}$$

Application in Channel Equalization

- Assuming n_k =0

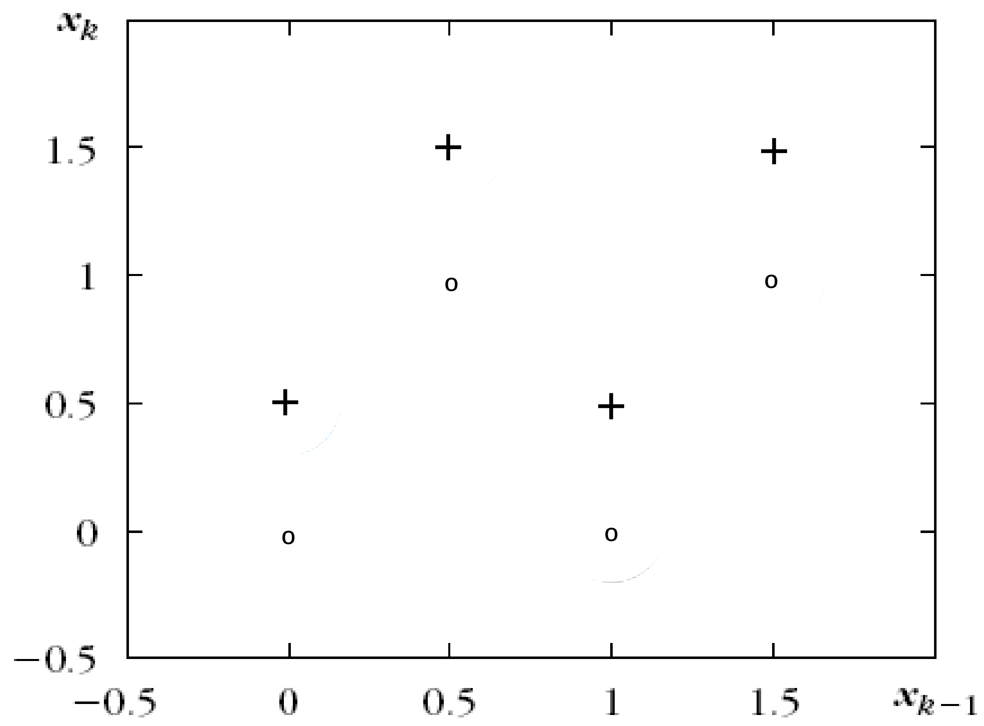
I_k	I_{k-1}	I_{k-2}	x_k	x_{k-1}	
0	0	0	0	0	
0	0	1	0	1	
0	1	0	1	0.5	
0	1	1	1	1.5	
1	0	0	0.5	0	
1	0	1	0.5	1	
1	1	0	1.5	0.5	
1	1	1	1.5	1.5	

- Estimate I_k from (x_k, x_{k-1})

Application in Channel Equalization

- Assuming n_k =0

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

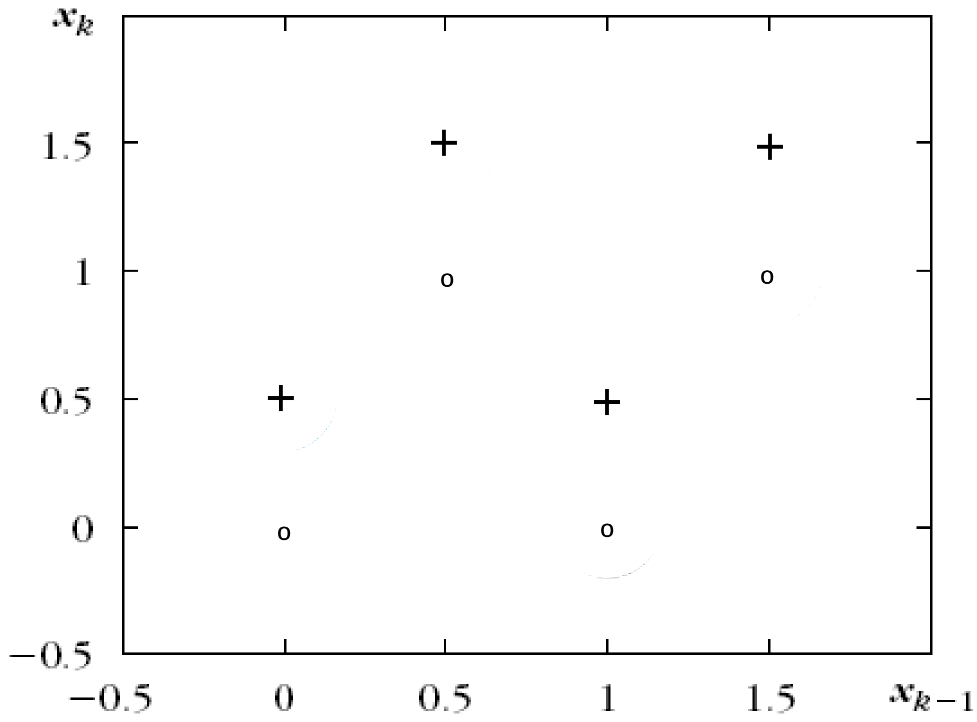


- Eight possible clusters

Application in Channel Equalization

- Assuming n_k =0

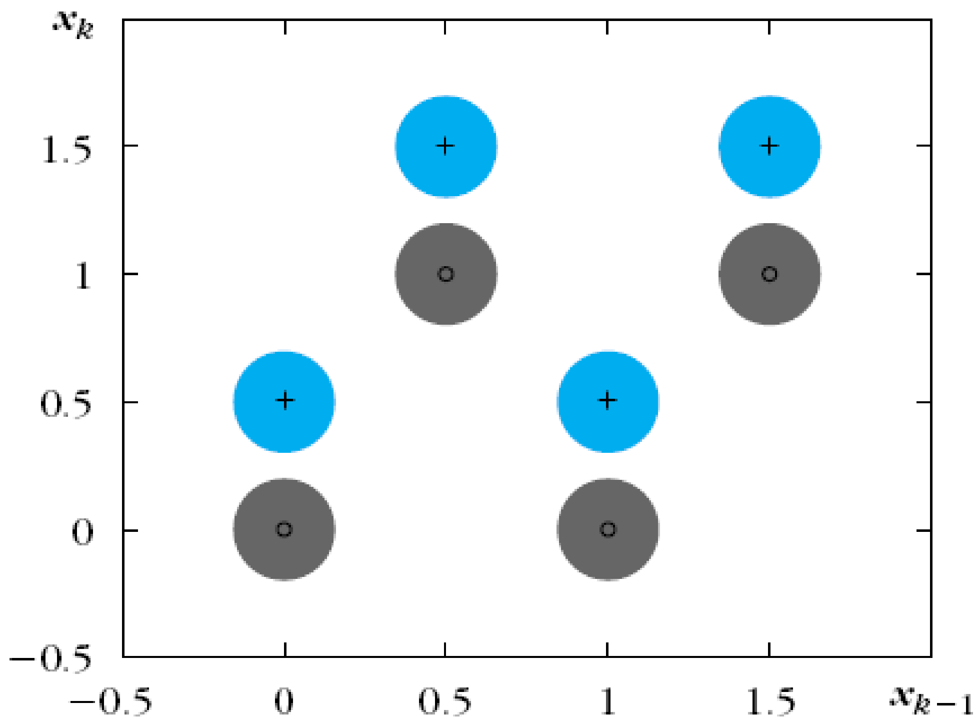
I_k	I_{k-1}	I_{k-2}	x_k	x_{k-1}	
0	0	0	0	0	ω_1
0	0	1	0	1	ω_2
0	1	0	1	0.5	ω_3
0	1	1	1	1.5	ω_4
1	0	0	0.5	0	ω_5
1	0	1	0.5	1	ω_6
1	1	0	1.5	0.5	ω_7
1	1	1	1.5	1.5	ω_8



- '+' means I_k was 1, 'o' means I_k was 0

Application in Channel Equalization

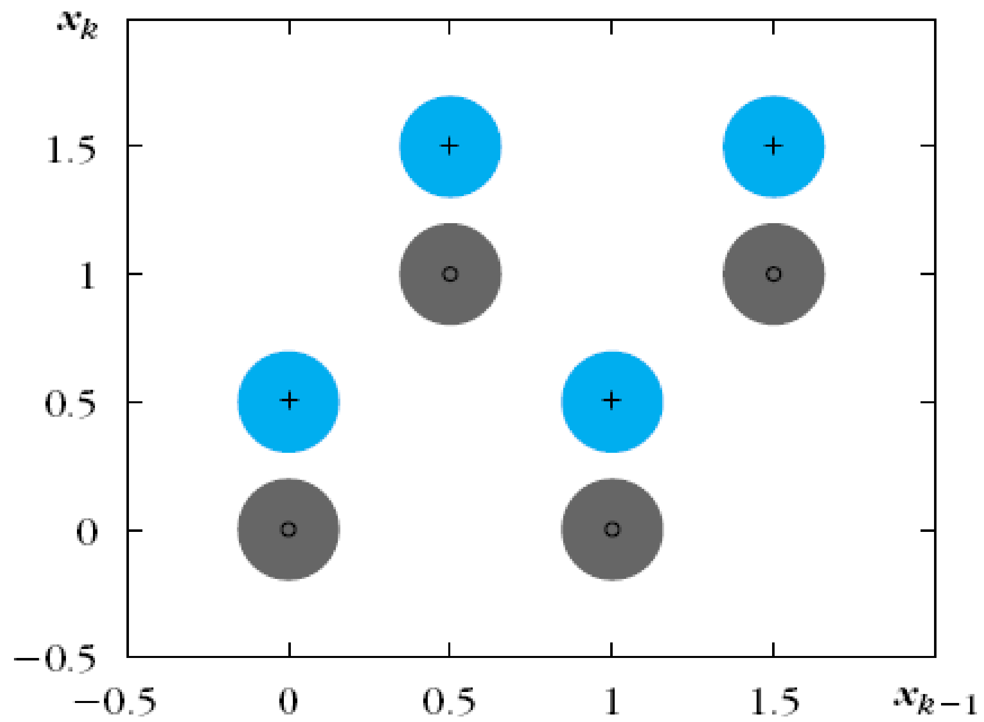
I_k	I_{k-1}	I_{k-2}	x_k	x_{k-1}	
0	0	0	0	0	ω_1
0	0	1	0	1	ω_2
0	1	0	1	0.5	ω_3
0	1	1	1	1.5	ω_4
1	0	0	0.5	0	ω_5
1	0	1	0.5	1	ω_6
1	1	0	1.5	0.5	ω_7
1	1	1	1.5	1.5	ω_8



- Big cluster exists when the noise n_k is NOT 0 (*zero*)

Application in Channel Equalization

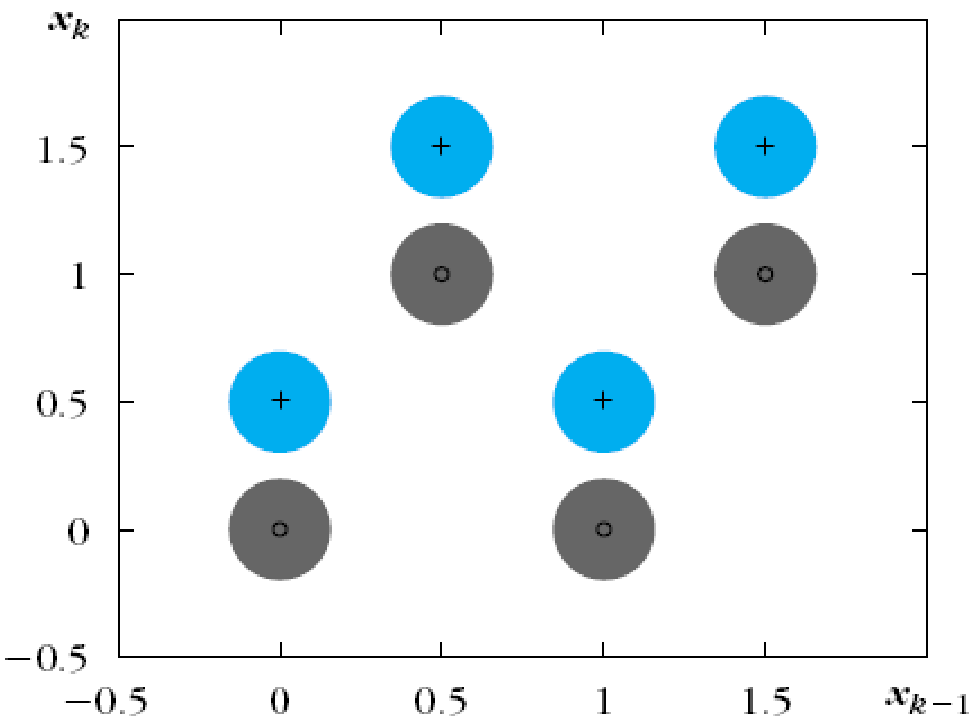
I_k	I_{k-1}	I_{k-2}	x_k	x_{k-1}	
0	0	0	0	0	ω_1
0	0	1	0	1	ω_2
0	1	0	1	0.5	ω_3
0	1	1	1	1.5	ω_4
1	0	0	0.5	0	ω_5
1	0	1	0.5	1	ω_6
1	1	0	1.5	0.5	ω_7
1	1	1	1.5	1.5	ω_8



- Estimate $\hat{I}_k$ from (x_k, x_{k-1})
- A two class problem
- Each class is a union of clusters

Solution (1)

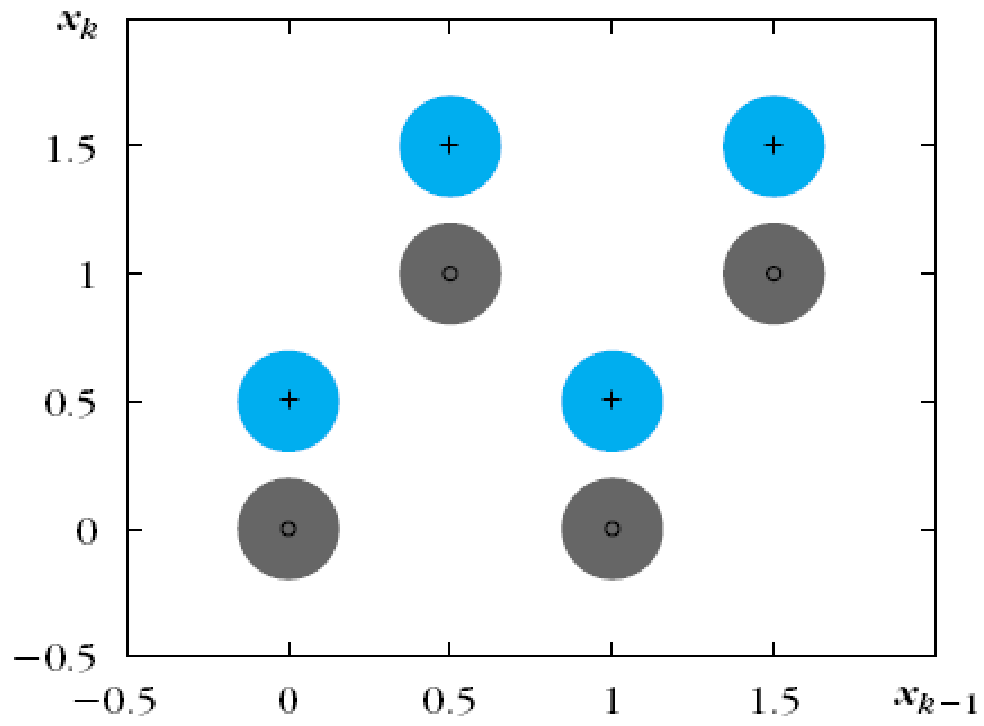
I_k	I_{k-1}	I_{k-2}	x_k	x_{k-1}	
0	0	0	0	0	ω_1
0	0	1	0	1	ω_2
0	1	0	1	0.5	ω_3
0	1	1	1	1.5	ω_4
1	0	0	0.5	0	ω_5
1	0	1	0.5	1	ω_6
1	1	0	1.5	0.5	ω_7
1	1	1	1.5	1.5	ω_8



- During training,
  - send all values of (I_{k-2}, I_{k-1}, I_k) and calculate (x_{k-1}, x_k)
  - Calculate the cluster centers μ_k
  - Assign the bit I_k to the cluster μ_k

Solution (1)

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



- During equalization (testing),
  - Get (x_{k-1}, x_k) from the channel
  - Find the nearest cluster μ_i
  - The bit of μ_i is the estimated transmitted bit

Solution (2)

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

- Let

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

which corresponds to ω_2

Solution (2)

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

- Let

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

which corresponds to ω_2

Solution (2)

I_{k+1}	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

- Let

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

which corresponds to ω_2

- What will be the next bit I_{k+1}

Solution (2)

I_{k+1}	I_k	I_{k-1}	I_{k-2}	x_k	x_{k-1}	
?	0	0	0	0	0	ω_1
0, 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

- Let

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

which corresponds to ω_2

- The next bit I_{k+1} can be either 1 or 0

$$(I_{k+1}, I_k, I_{k-1}) \begin{cases} (1, 0, 0) & \boxed{\omega_5} \\ (0, 0, 0) & \boxed{\omega_1} \end{cases}$$

Solution (2)

I_{k+1}	I_k	I_{k-1}	I_{k-2}	x_k	x_{k-1}	
?	0	0	0	0	0	ω_1
0, 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

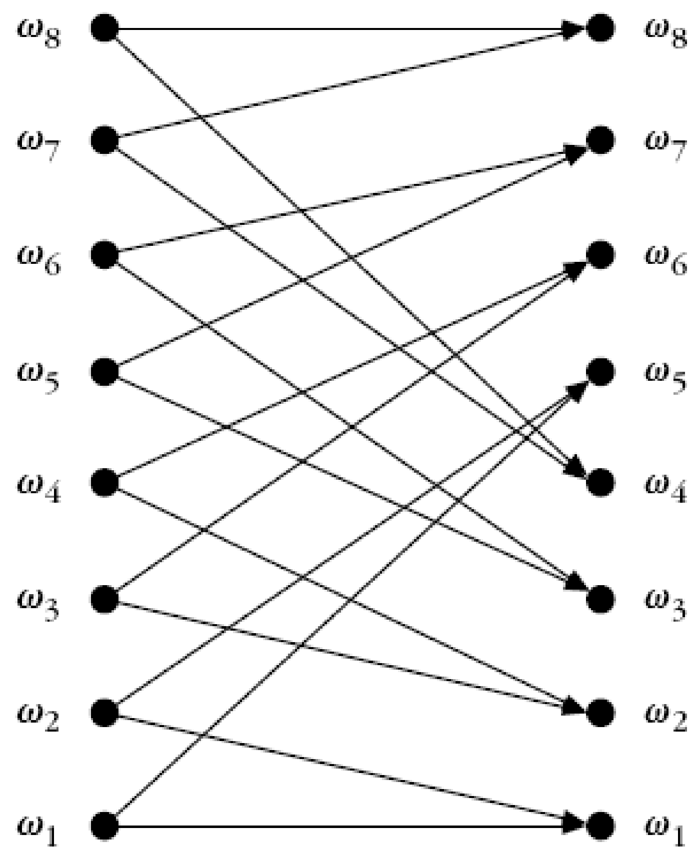
- This means, all transitions are not possible!!
- In other words, after (0, 0, 1) we will find

  either (1, 0, 0)

  or (0, 0, 0)

Solution (2)

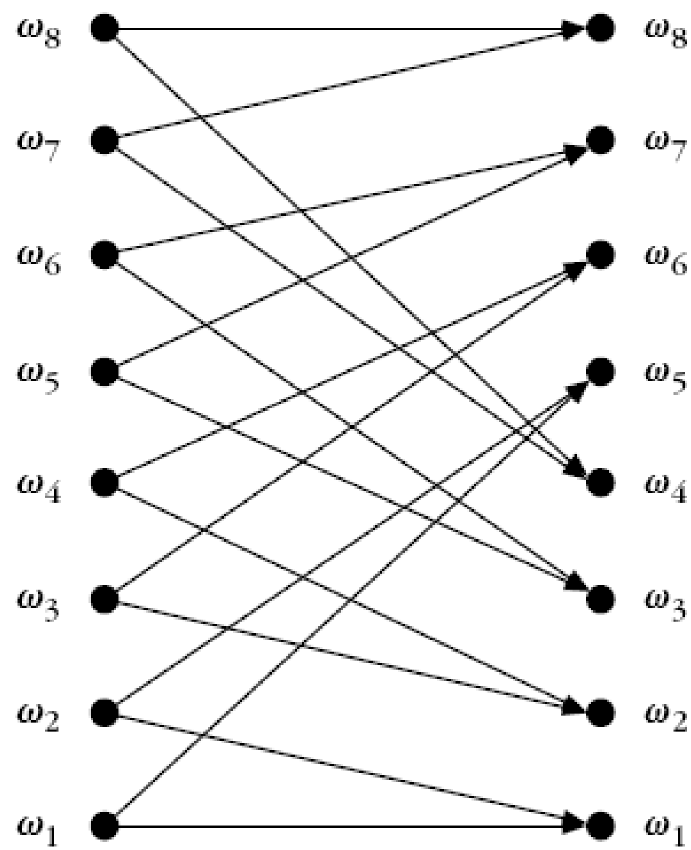
I_{k+1}	I_k	I_{k-1}	I_{k-2}	x_k	x_{k-1}	
0, 1	0	0	0	0	0	ω_1
0, 1	0	0	1	0	1	ω_2
0, 1	0	1	0	1	0.5	ω_3
0, 1	0	1	1	1	1.5	ω_4
0, 1	1	0	0	0.5	0	ω_5
0, 1	1	0	1	0.5	1	ω_6
0, 1	1	1	0	1.5	0.5	ω_7
0, 1	1	1	1	1.5	1.5	ω_8



- This means, all transitions are NOT possible!!
- Alternately, after (0, 0, 1) we can get either (1, 0, 0) or (0, 0,0)

Solution (2)

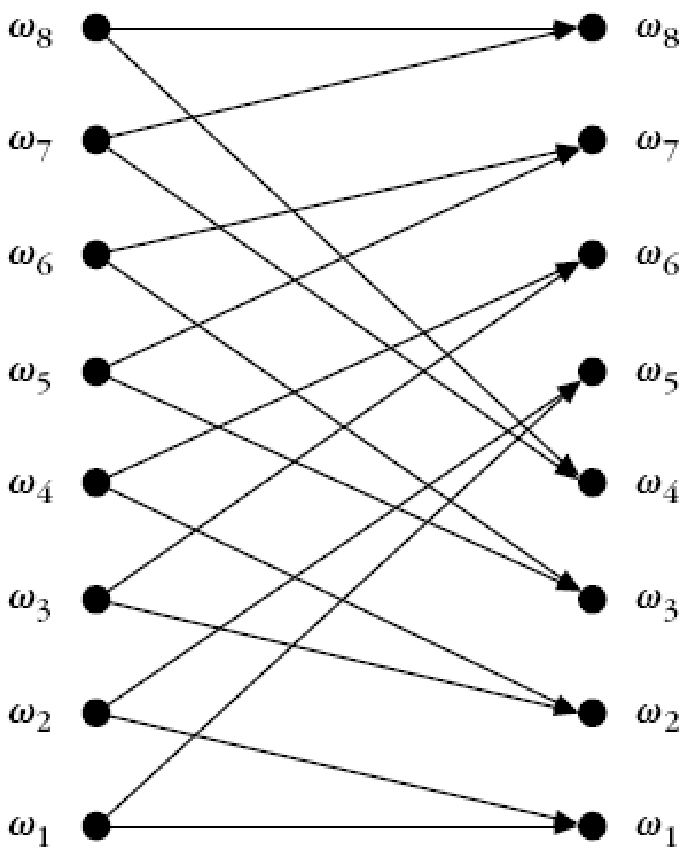
I_{k+1}	I_k	I_{k-1}	I_{k-2}	x_k	x_{k-1}	
0, 1	0	0	0	0	0	ω_1
0, 1	0	0	1	0	1	ω_2
0, 1	0	1	0	1	0.5	ω_3
0, 1	0	1	1	1	1.5	ω_4
0, 1	1	0	0	0.5	0	ω_5
0, 1	1	0	1	0.5	1	ω_6
0, 1	1	1	0	1.5	0.5	ω_7
0, 1	1	1	1	1.5	1.5	ω_8



- We can use Viterbi algorithm now !!!

Solution (2): Viterbi Algorithm

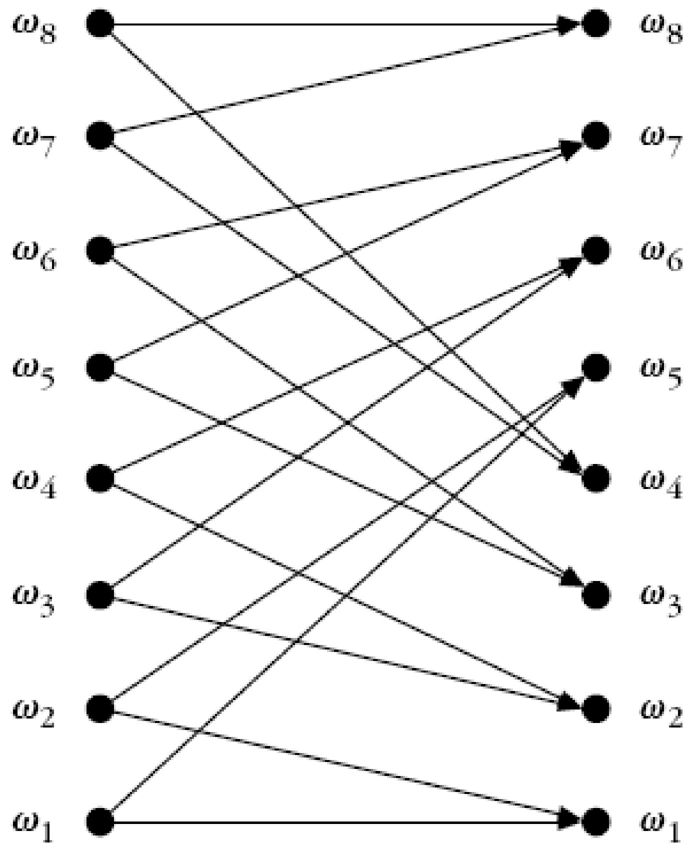
I_{k+1}	I_k	I_{k-1}	I_{k-2}	x_k	x_{k-1}	
0, 1	0	0	0	0	0	ω_1
0, 1	0	0	1	0	1	ω_2
0, 1	0	1	0	1	0.5	ω_3
0, 1	0	1	1	1	1.5	ω_4
0, 1	1	0	0	0.5	0	ω_5
0, 1	1	0	1	0.5	1	ω_6
0, 1	1	1	0	1.5	0.5	ω_7
0, 1	1	1	1	1.5	1.5	ω_8



- We need to define:
- $P(\omega_i | \omega_j)$
- Cost function

Solution (2): Viterbi Algorithm

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



- Assuming equiprobability of the next bit:
  - $P(\omega_1 | \omega_1) = 0.5 = P(\omega_1 | \omega_5)$

Solution (2): Viterbi Algorithm

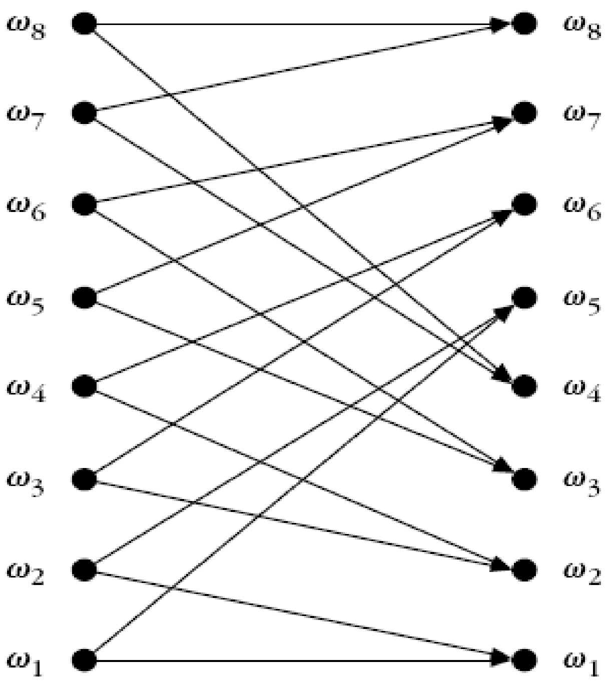
- Cost function of transition

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

Where, either Euclidean distance

$$d_{\omega_{i_k}}(\boldsymbol{x}_k) = \|\boldsymbol{x}_k - \boldsymbol{\mu}_{i_k}\|$$

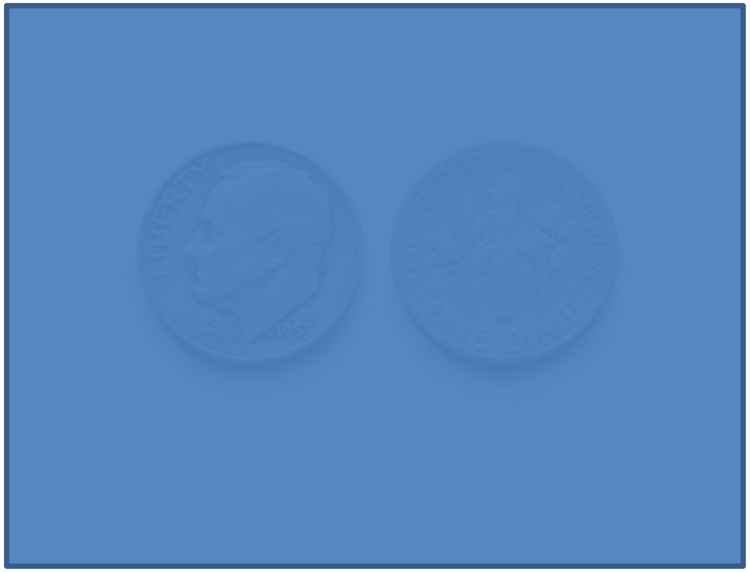
Or Mahalanobis distance



$$d_{\omega_{i_k}}(\boldsymbol{x}_k) = \left((\boldsymbol{x}_k - \boldsymbol{\mu}_{i_k})^T \Sigma_{i_k}^{-1} (\boldsymbol{x}_k - \boldsymbol{\mu}_{i_k})\right)^{1/2}$$

Can be used

Hidden Markov Model



Markov Models

- Set of states: $\{s_1, s_2, \ldots, s_N\}$

Markov Models

- Set of states: $\{s_1, s_2, \ldots, s_N\}$
- Process moves from one state to another generating a sequence of states : $s_{i1}, s_{i2}, \ldots, s_{ik}, \ldots$

Markov Models

- Set of states: $\{s_1, s_2, \ldots, s_N\}$
- Process moves from one state to another generating a sequence of states : $s_{i1}, s_{i2}, \ldots, s_{ik}, \ldots$

- Markov chain property: probability of each subsequent state depends only on what was the previous state:

$$P(s_{ik} \mid s_{i1}, s_{i2}, \ldots, s_{ik-1}) = P(s_{ik} \mid s_{ik-1})$$

Markov Models

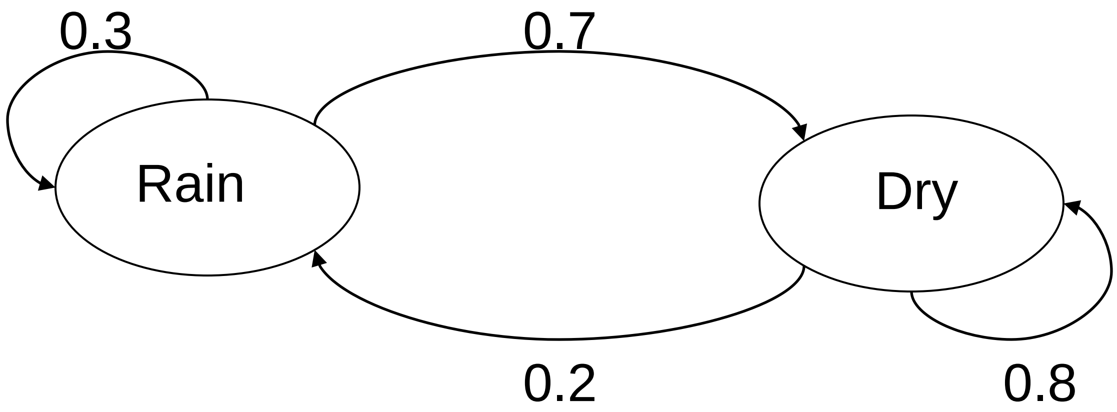
- Set of states: $\{s_1, s_2, \ldots, s_N\}$
- Process moves from one state to another generating a sequence of states : $s_{i1}, s_{i2}, \ldots, s_{ik}, \ldots$

- Markov chain property: probability of each subsequent state depends only on what was the previous state:

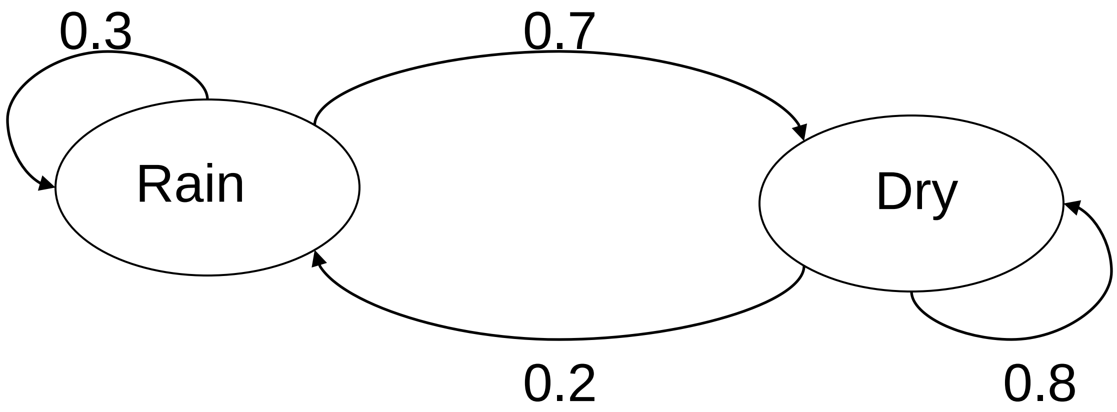
$$P(s_{ik} \mid s_{i1}, s_{i2}, \ldots, s_{ik-1}) = P(s_{ik} \mid s_{ik-1})$$

- To define Markov model, the following probabilities have to be specified: transition probabilities $a_{ij} = P(s_j \mid s_i)$ and initial probabilities $\pi_i = P(s_i)$

Example of Markov Model

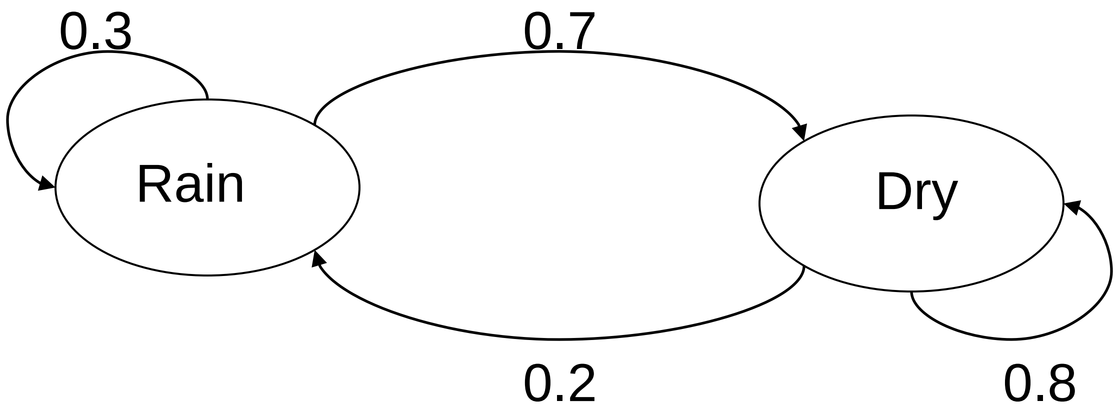


Example of Markov Model



- Two states : ‘Rain’ and ‘Dry’.

Example of Markov Model

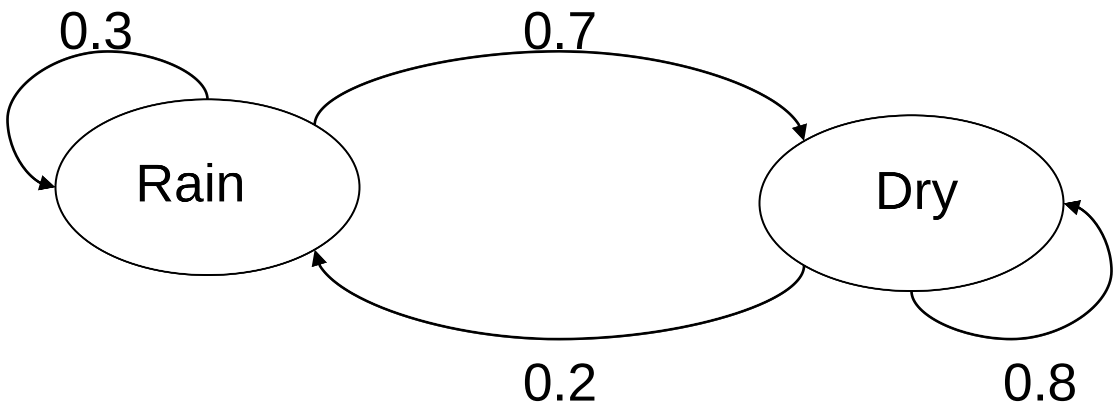


- Two states : ‘Rain’ and ‘Dry’.

•Transition probabilities:

  •P(‘Rain’|‘Rain’)=0.3 , P(‘Dry’|‘Rain’)=0.7 ,
  • P(‘Rain’|‘Dry’)=0.2, P(‘Dry’|‘Dry’)=0.8

Example of Markov Model



- Two states : ‘Rain’ and ‘Dry’.

- Transition probabilities:

  - P(‘Rain’|‘Rain’)=0.3 , P(‘Dry’|‘Rain’)=0.7 ,
  - P(‘Rain’|‘Dry’)=0.2, P(‘Dry’|‘Dry’)=0.8

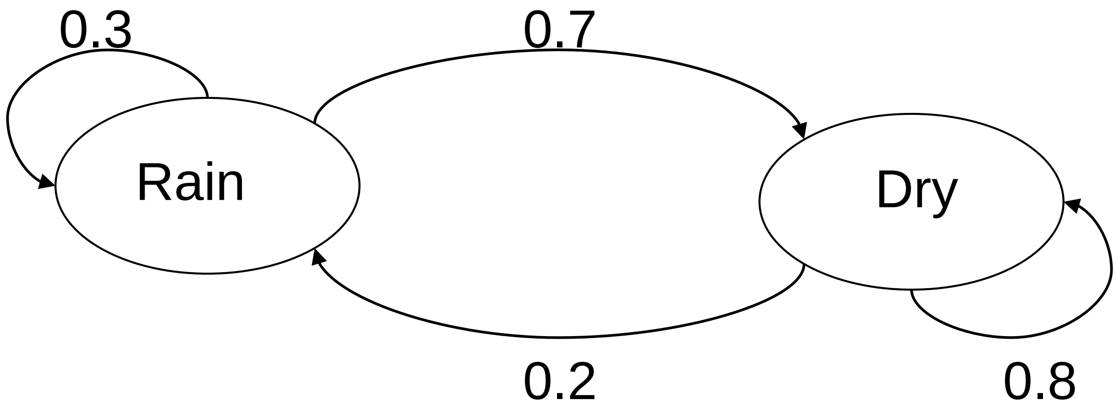
- Initial probabilities: say P(‘Rain’)=0.4 , P(‘Dry’)=0.6 .

Calculation of sequence probability

- By Markov chain property, probability of state sequence can be found by the formula:

$$P(s_{i1}, s_{i2}, \ldots, s_{ik}) = P(s_{ik} \mid s_{i1}, s_{i2}, \ldots, s_{ik-1}) P(s_{i1}, s_{i2}, \ldots, s_{ik-1})$$
$$= P(s_{ik} \mid s_{ik-1}) P(s_{i1}, s_{i2}, \ldots, s_{ik-1}) = \ldots$$
$$= P(s_{ik} \mid s_{ik-1}) P(s_{ik-1} \mid s_{ik-2}) \ldots P(s_{i2} \mid s_{i1}) P(s_{i1})$$

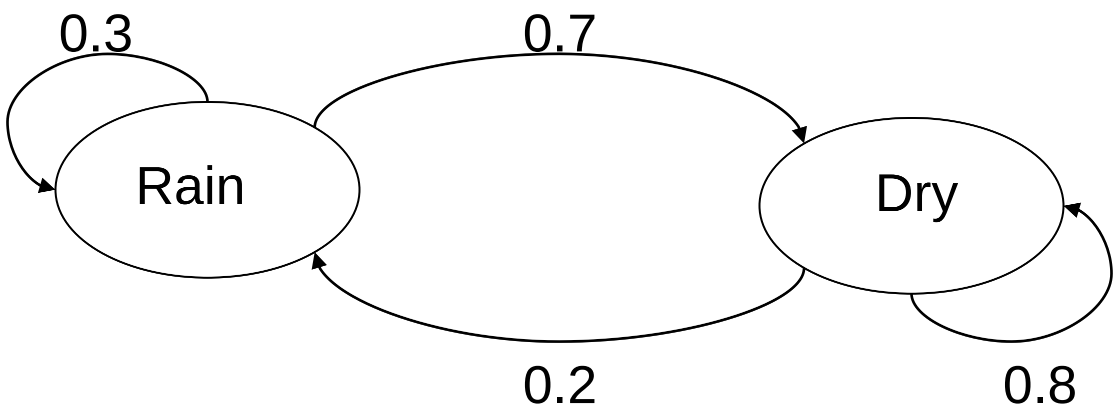
Calculation of sequence probability



•Suppose we want to calculate a probability of a sequence of states in our example, {‘Dry’,’Dry’,’Rain’,Rain’}.

P({‘Dry’,’Dry’,’Rain’,Rain’})

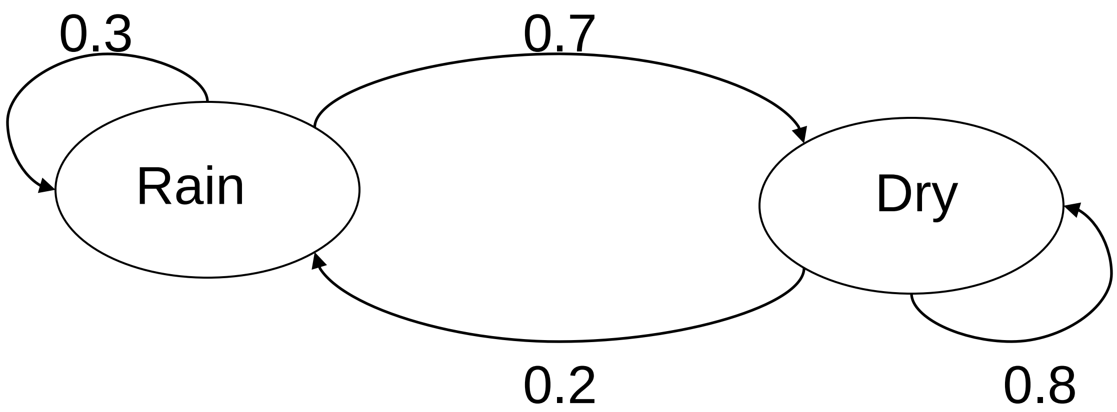
Calculation of sequence probability



•Suppose we want to calculate a probability of a sequence of states in our example, {'Dry','Dry','Rain',Rain'}.

P({'Dry','Dry','Rain',Rain'})
= P('Rain'|'Rain') P('Rain'|'Dry') P('Dry'|'Dry') P('Dry')

Calculation of sequence probability



•Suppose we want to calculate a probability of a sequence of states in our example, {'Dry','Dry','Rain',Rain'}.

P({'Dry','Dry','Rain',Rain'})

= P('Rain'|'Rain') P('Rain'|'Dry') P('Dry'|'Dry') P('Dry')

= 0.3*0.2*0.8*0.6

Hidden Markov Model

Hidden Markov Model

Hidden Markov Model

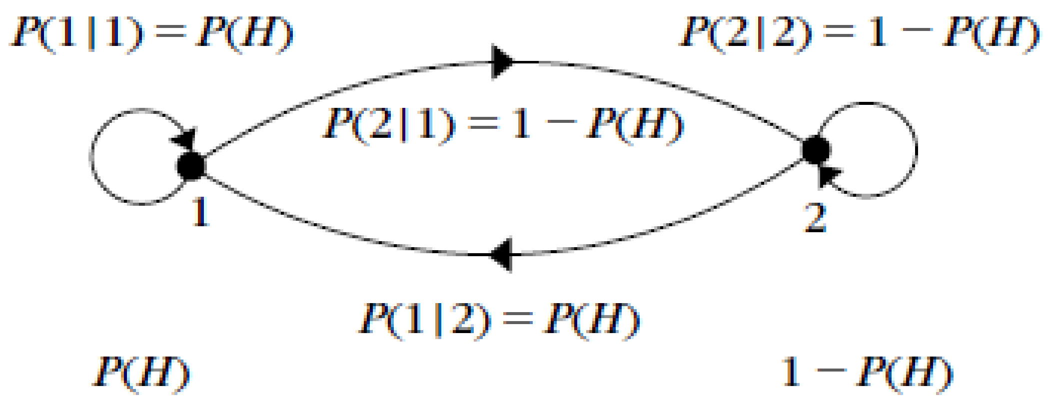
H T H H T T T H H H . . . . .

Hidden Markov Model

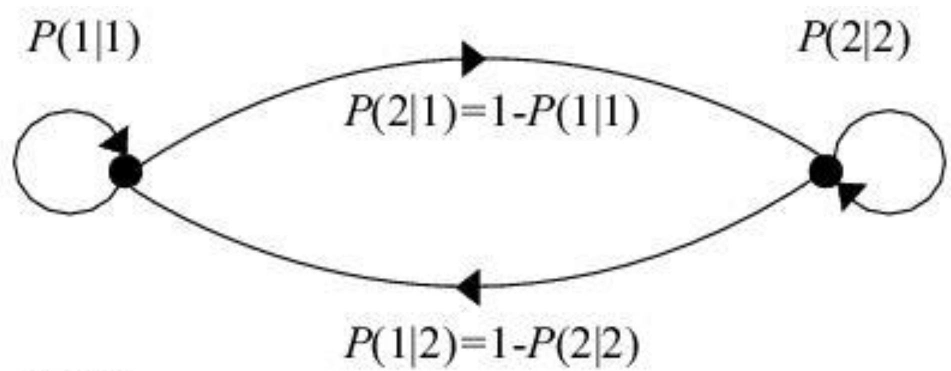
H T H H T T T H H H . . . . .

Can we guess which coin is tossed at different times?

Not A Hidden Markov Model



Hidden Markov Model



$P_1(H)$

$P_1(T)=1-P_1(H)$

$P_2(H)$

$P_2(T)=1-P_2$

Hidden Markov models

- Set of states: $\{s_1, s_2, \ldots, s_N\}$

•Process moves from one state to another generating a sequence of states : $s_{i1}, s_{i2}, \ldots, s_{ik}, \ldots$

•Markov chain property: probability of each subsequent state depends only on what was the previous state:

$$P(s_{ik} \mid s_{i1}, s_{i2}, \ldots, s_{ik-1}) = P(s_{ik} \mid s_{ik-1})$$

Hidden Markov models

- States are not visible, but each state randomly generates one of M observations (or visible states)

$$\{v_1, v_2, \ldots, v_M\}$$

Hidden Markov models

• States are not visible, but each state randomly generates one of *M* observations (or visible states)

$$\{v_1, v_2, \ldots, v_M\}$$

•To define hidden Markov model, the following probabilities have to be specified:

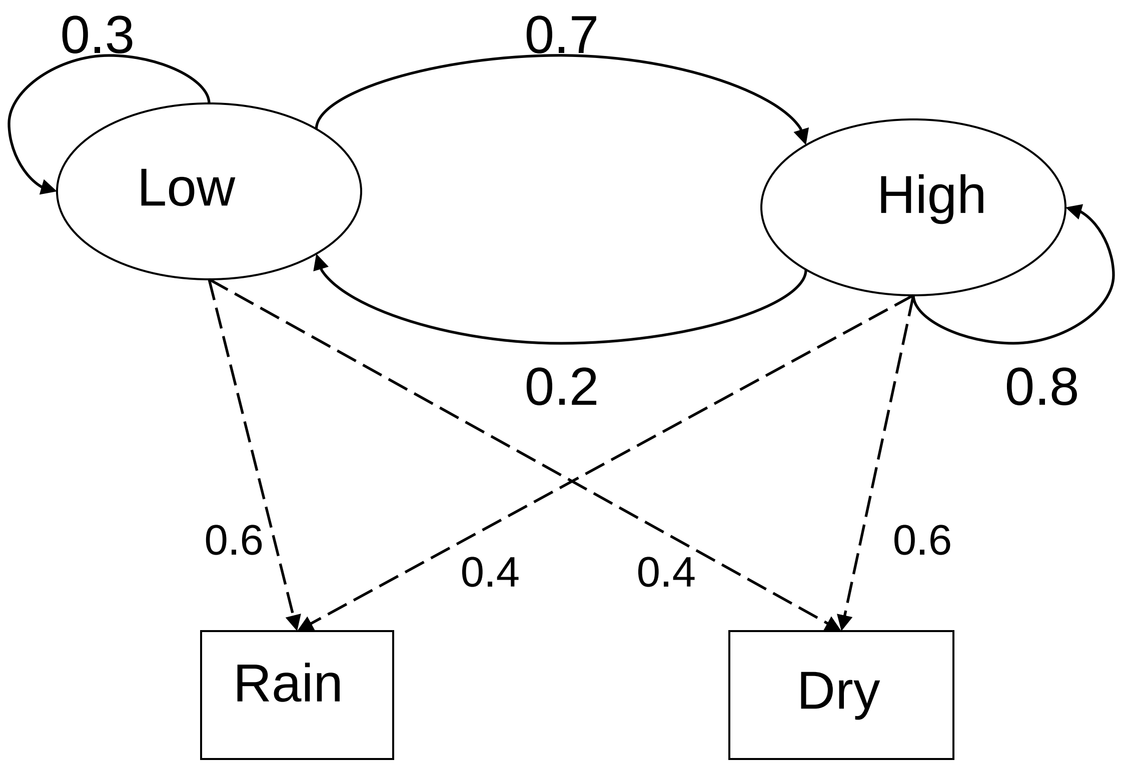
•matrix of transition probabilities $A=(a_{ij})$, $a_{ij}= P(s_j \mid s_i)$

•matrix of observation probabilities $B=(b_i(v_m))$,
where, $b_i(v_m) = P(v_m|s_i)$

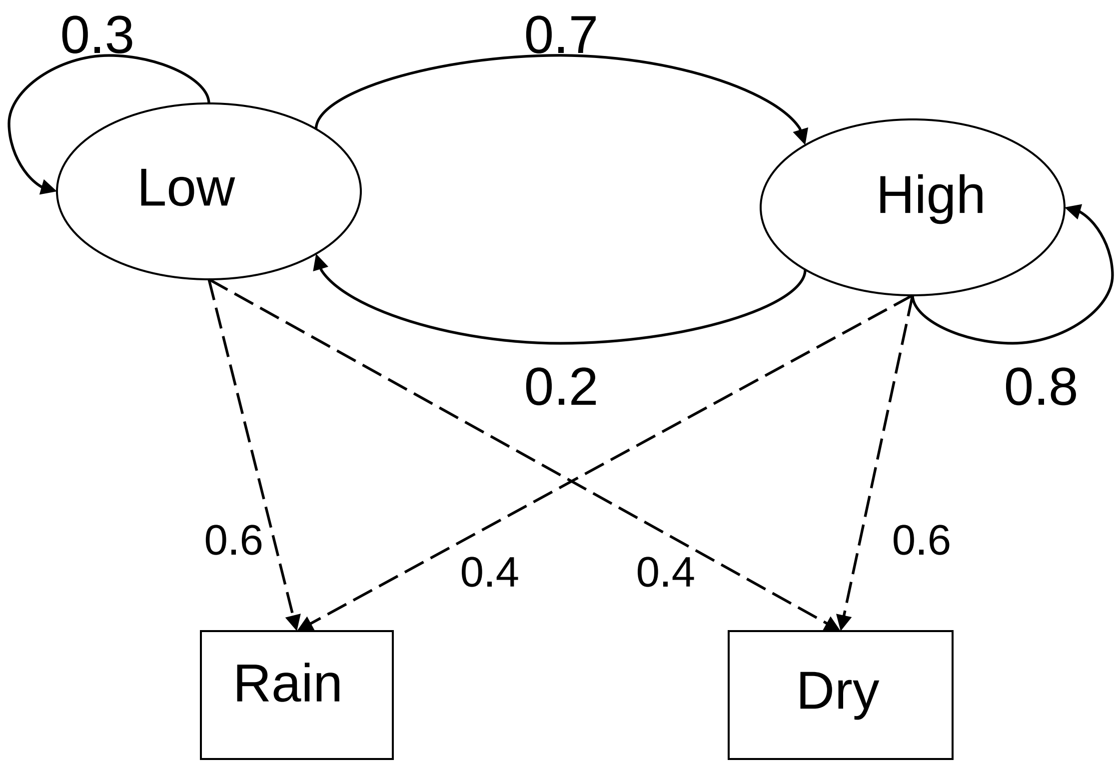
•vector of initial probabilities $\pi=(\pi_i)$, $\pi_i = P(s_i)$

•Model is represented by $M=(A, B, \pi)$.

Example of Hidden Markov Model



Example of Hidden Markov Model



•Initial probabilities: say P('Low')=0.4 , P('High')=0.6 .

Example of Hidden Markov Model

- Two states : ‘Low’ and ‘High’ atmospheric pressure.
- Two observations : ‘Rain’ and ‘Dry’.
- Transition probabilities: $P(\text{‘Low’}|\text{‘Low’})=0.3$, $P(\text{‘High’}|\text{‘Low’})=0.7$, $P(\text{‘Low’}|\text{‘High’})=0.2$, $P(\text{‘High’}|\text{‘High’})=0.8$
- Observation probabilities : $P(\text{‘Rain’}|\text{‘Low’})=0.6$, $P(\text{‘Dry’}|\text{‘Low’})=0.4$, $P(\text{‘Rain’}|\text{‘High’})=0.4$, $P(\text{‘Dry’}|\text{‘High’})=0.3$.
- Initial probabilities: say $P(\text{‘Low’})=0.4$, $P(\text{‘High’})=0.6$.

Calculation of observation sequence probability

- Suppose we want to calculate a probability of a sequence of observations in our example, {'Dry','Rain'}.

- Consider all possible hidden state sequences:

P({'Dry','Rain'}) = P({'Dry','Rain'} , {'Low','Low'})
+P({'Dry','Rain'} , {'Low','High'})
+ P({'Dry','Rain'} , {'High','Low'})
+ P({'Dry','Rain'} , {'High','High'})

Calculation of observation sequence probability

- P({'Dry','Rain'}) = P({'Dry','Rain'} , {'Low','Low'})
  + P({'Dry','Rain'} , {'Low','High'})
  + P({'Dry','Rain'} , {'High','Low'})
  + P({'Dry','Rain'} , {'High','High'})

where first term is :
P({'Dry','Rain'} , {'Low','Low'})=
P({'Dry','Rain'} | {'Low','Low'}) P({'Low','Low'}) =
P('Dry'|'Low')P('Rain'|'Low') P('Low')P('Low'|'Low)
= 0.4*0.4*0.6*0.4*0.3

Main issues using HMMs

- **Evaluation problem.**

Given the HMM $M=(A, B, \pi)$ and the observation sequence $O=o_1\, o_2 \ldots o_K$, calculate the probability that model M has generated sequence O .

$O=o_1 \ldots o_K$ denotes a sequence of observations $o_k \in \{v_1, \ldots, v_M\}$.

Main issues using HMMs

• **Decoding problem.**

Given the HMM $M=(A, B, \pi)$ and the observation sequence $O=o_1\, o_2 \ldots o_K$, calculate the most likely sequence of hidden states s_i that produced this observation sequence O.

$O=o_1 \ldots o_K$ denotes a sequence of observations $o_k \in \{v_1, \ldots, v_M\}$.

Main issues using HMMs

•**Learning problem.**

Given some training observation sequences $O=o_1\, o_2 \ldots o_K$ and general structure of HMM (numbers of hidden and visible states), determine HMM parameters $M=(A, B, \pi)$ that best fit training data.

$O=o_1\ldots o_K$ denotes a sequence of observations $o_k \in \{v_1,\ldots,v_M\}$.