Machine Learning
Hidden Markov Models

Mannes Poel
Gwenn Englebienne

Introduction

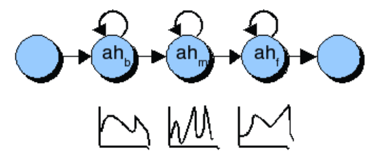
- Modeling dependencies in input; no longer iid
- Sequences:
  - ***Temporal:***
    - ***In speech; phonemes in a word (dictionary), words in a sentence (syntax, semantics of the language).***
    - ***In handwriting, pen movements, gesture recognition.***
    - ***In brain signals; temporal aspects such as a P300***
  - ***Spatial: In a DNA sequence; base pairs***

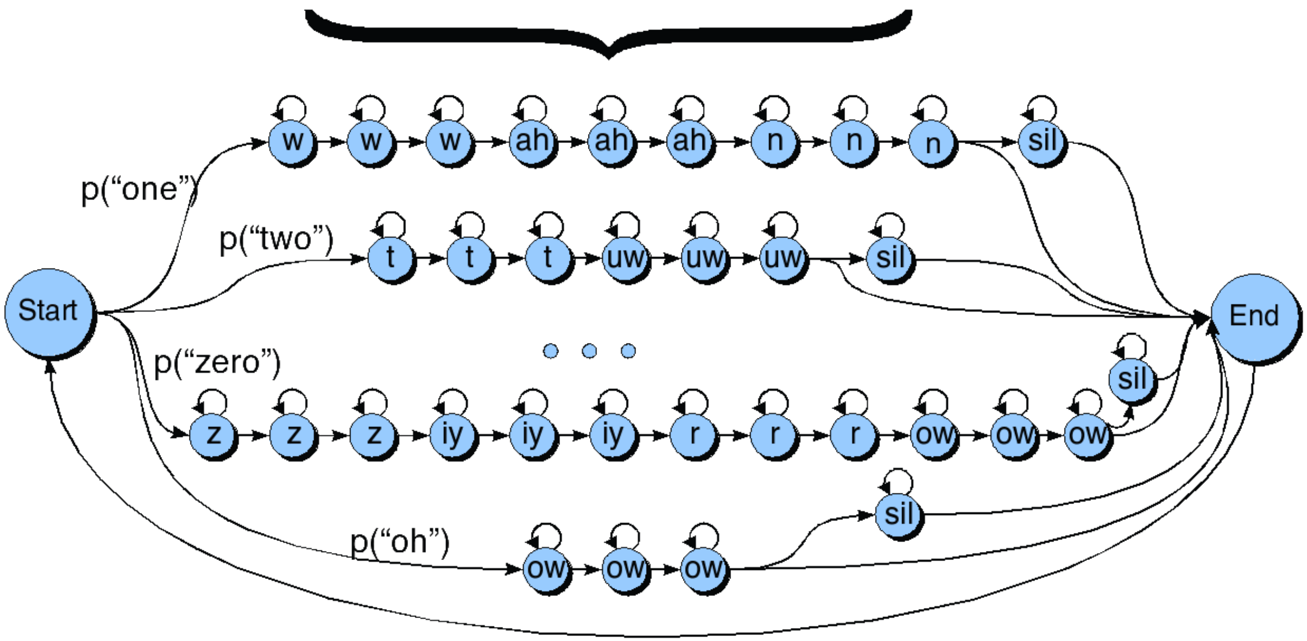
Speech detection

Lexicon

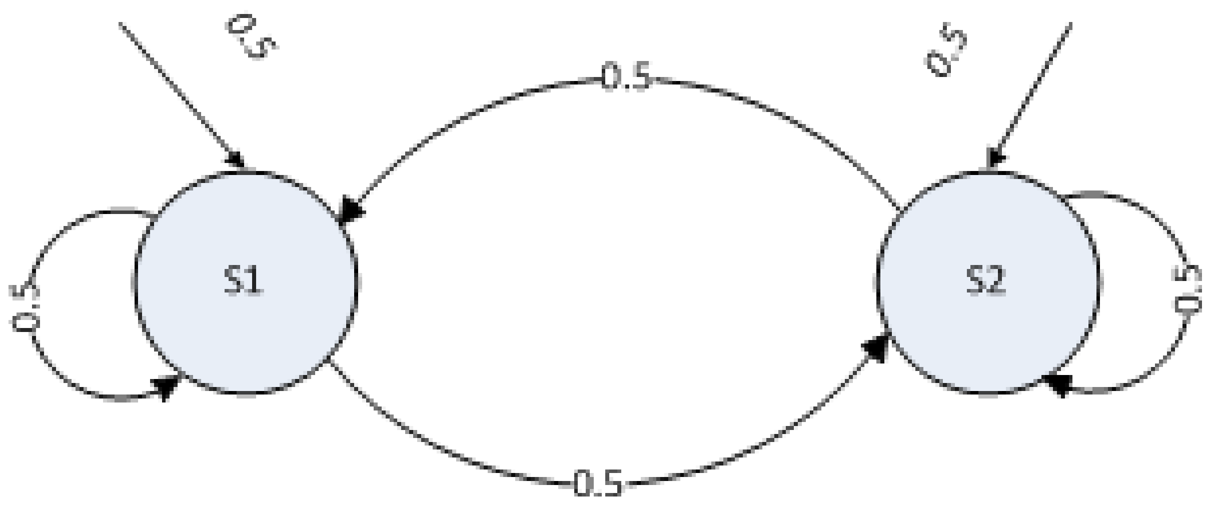
one	w ah n
two	t uw
three	th r iy
four	f ao r
five	f ay v
six	s ih k s
seven	s eh v ax n
eight	ey t
nine	n ay n
zero	z iy r ow
oh	ow

Phone HMM





Probabilistic Transition Models, Stochastic Automaton, Markov Models



- Gives rise to state sequences of the form:

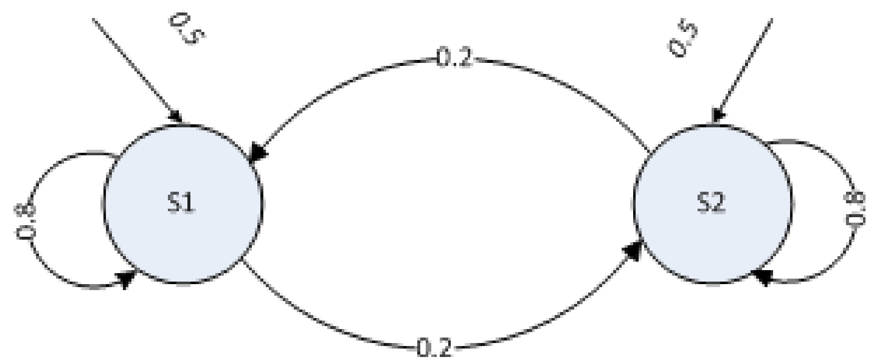
S1S1S2S2S1S1S2S1

Quiz

Given the following state sequence ***s***

s=S1S2S1S2S1S2S1S2

and stochastic automaton ***M***

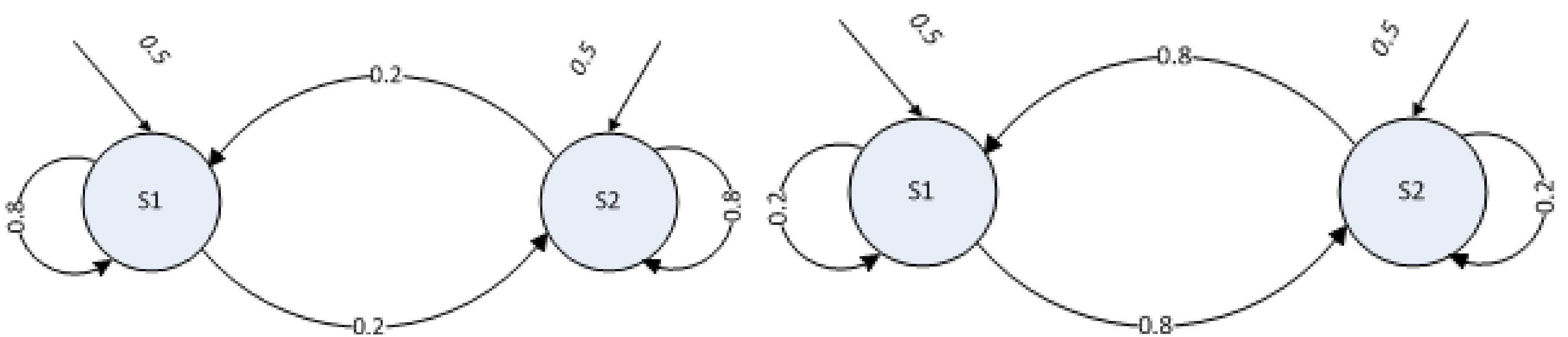


What is the likelihood that ***s*** is generated by ***M;*** how to compute ***P(s|M)***?

Quiz

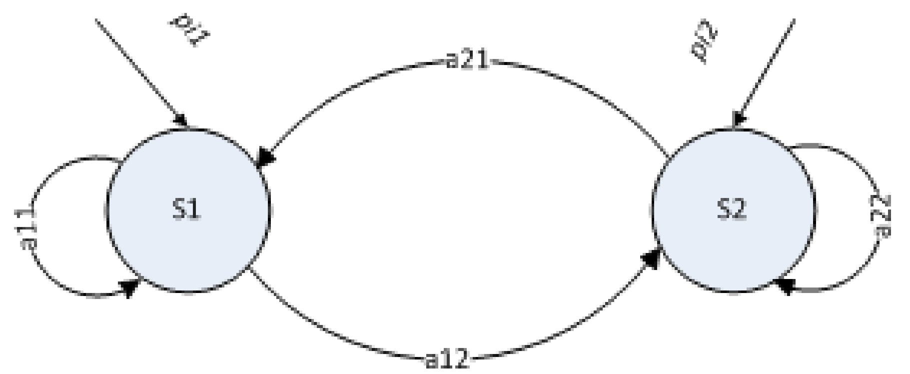
- Given the following state sequence ***s*** which model is more likely to generate such a sequence; i.e. maxarg_M ***P(s|M)?***

s=S1S2S1S2S1S2S1S2



Quiz

- Given the sequences:
  - S1S2S2S1S1S1S2S2S1S1S2
  - S1S1S1S2S2S1S1S2S2S1S1
  - S2S2S2S1S1S2S1S2S2S1S1
- How to estimate the probabilities?



Discrete Markov Process

- $\boldsymbol{N}$ states: $\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_{\boldsymbol{N}}$
  - State at "time" $\boldsymbol{t}$, $\boldsymbol{q}_{\boldsymbol{t}} = \boldsymbol{S}_{\boldsymbol{i}}$
- First-order Markov

$$P(q_{t+1}=S_j \mid q_t=S_i, q_{t-1}=S_k, \ldots) = P(q_{t+1}=S_j \mid q_t=S_i)$$

- Transition probabilities

$$a_{ij} \equiv P(q_{t+1}=S_j \mid q_t=S_i) \qquad a_{ij} \geq 0 \text{ and } \Sigma_{j=1}^{N} a_{ij}=1$$

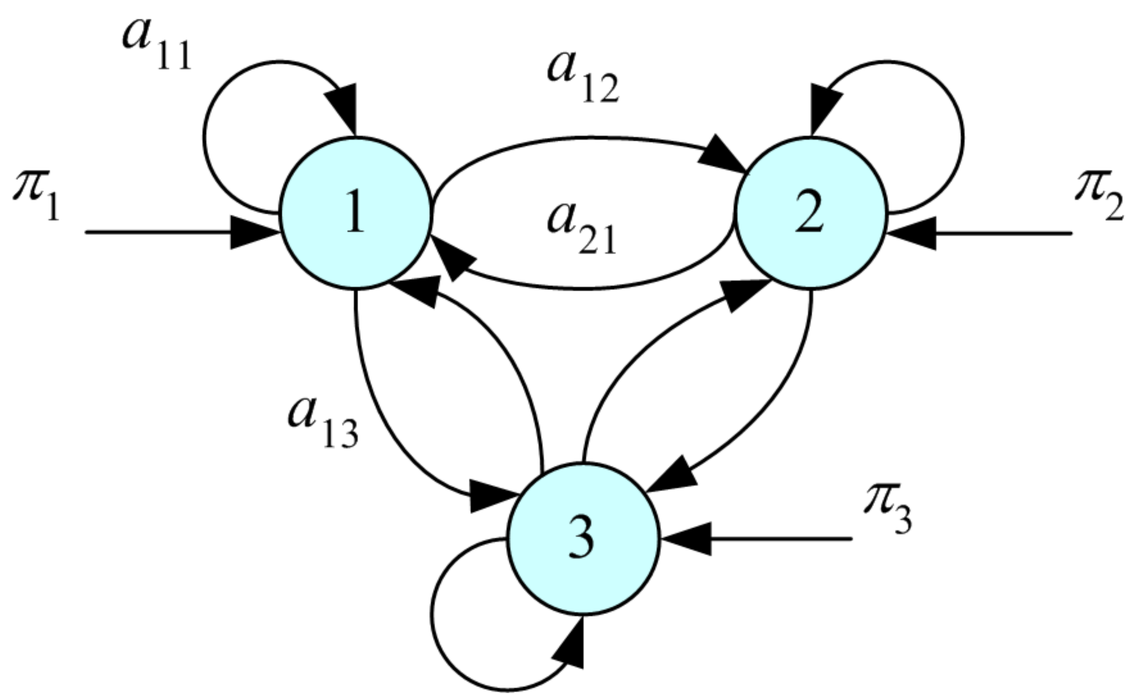
- Initial probabilities

$$\pi_i \equiv P(q_1=S_i) \qquad \Sigma_{j=1}^{N} \pi_i=1$$

Example of 2 coins

- Recall the estimation of coin probabilities. Can be modeled as an Hidden Markov Model

Stochastic Automaton



$$P(O = Q \mid \mathbf{A}, \mathbf{\Pi}) = P(q_1) \prod_{t=2}^{T} P(q_t \mid q_{t-1}) = \pi_{q_1} a_{q_1 q_2} \cdots a_{q_{T-1} q_T}$$

Example: Balls and Urns

- Three urns each full of balls of one color

  $\boldsymbol{S}_1$: red, $\boldsymbol{S}_2$: blue, $\boldsymbol{S}_3$: green

$$\Pi = [0.5, 0.2, 0.3]^T \qquad \mathbf{A} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

$$O = \{S_1, S_1, S_3, S_3\}$$

$$P(O \mid \mathbf{A}, \Pi) = P(S_1) \cdot P(S_1 \mid S_1) \cdot P(S_3 \mid S_1) \cdot P(S_3 \mid S_3)$$

$$= \pi_1 \cdot a_{11} \cdot a_{13} \cdot a_{33}$$

$$= 0.5 \cdot 0.4 \cdot 0.3 \cdot 0.8 = 0.048$$

Balls and Urns: Learning

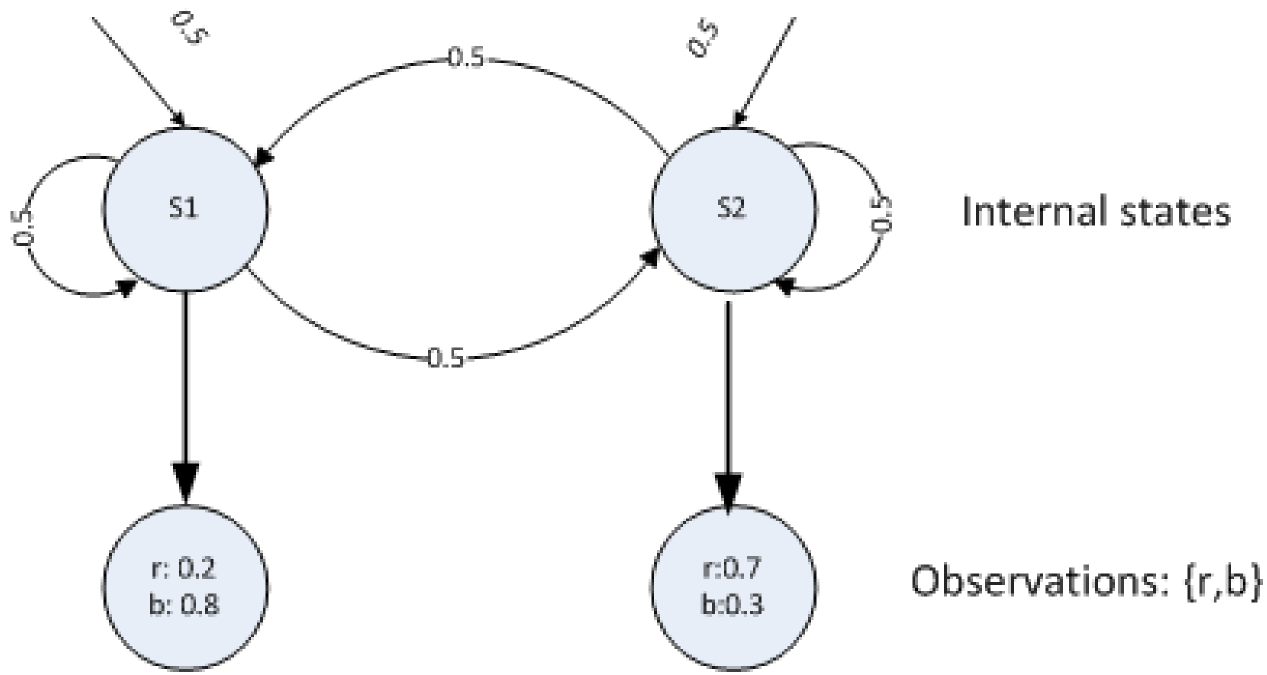
- Given ***K*** example sequences of length ***T***

$$\hat{\pi}_i = \frac{\#\{\text{sequences starting with } S_i\}}{\#\{\text{sequences}\}} = \frac{\sum_k 1\left(q_1^k = S_i\right)}{K}$$

$$\hat{a}_{ij} = \frac{\#\{\text{transition s from } S_i \text{ to } S_j\}}{\#\{\text{transition s from } S_i\}}$$

$$= \frac{\sum_k \sum_{t=1}^{T\text{-}1} 1\left(q_t^k = S_i \text{ and } q_{t+1}^k = S_j\right)}{\sum_k \sum_{t=1}^{T\text{-}1} 1\left(q_t^k = S_i\right)}$$

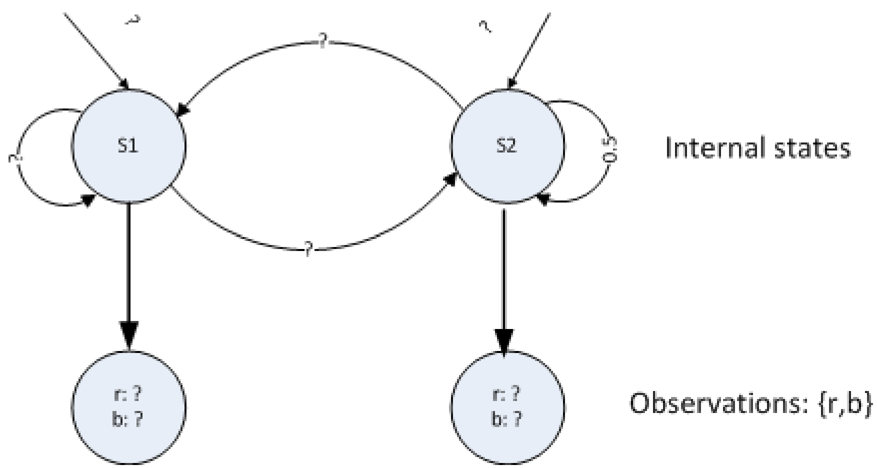
Hidden Markov Models



- Quiz: How to compute the likelihood that the sequence ***rrrbbrbbbbrb*** is generated by this HMM?

Quiz

- Given the sequences:
  - rrbbrrrbbrbrbrb
  - bbbbrrrrbbrbbbbrrbr
  - rbrbrrbbrbrbrbrbrbbbr
- ***How to estimate the probabilities?***



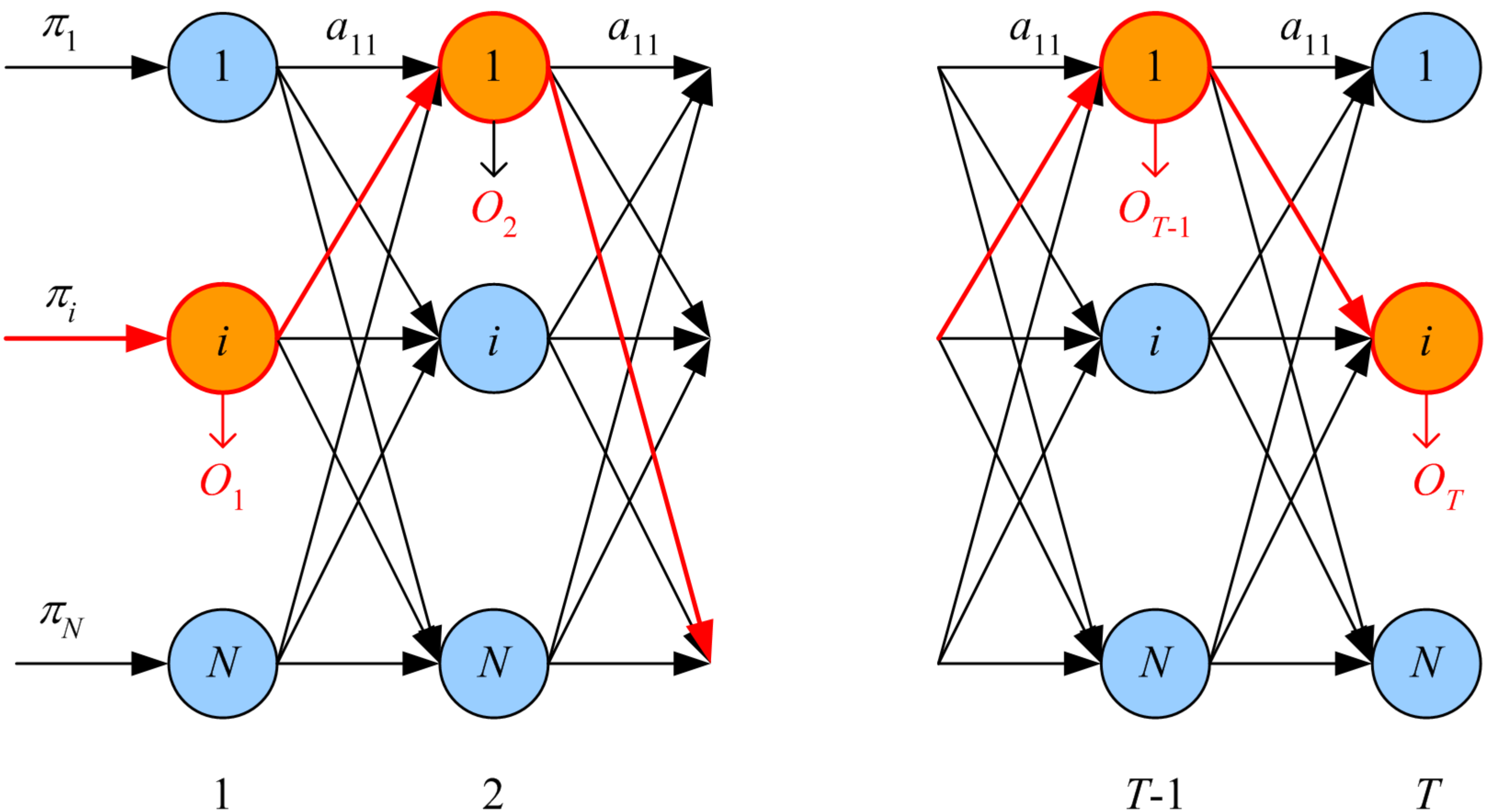
Hidden Markov Models

- States are not observable
- Discrete observations $\{\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_M\}$ are recorded; a probabilistic function of the state
- Emission probabilities

$$\boldsymbol{b_j(m)} \equiv \boldsymbol{P(O_t=v_m \mid q_t=S_j)}$$

- Example: In each urn, there are balls of different colors, but with different probabilities.
- For each observation sequence, there are multiple state sequences

HMM Unfolded in Time



Elements of an HMM

- ***N***: Number of states
- ***M***: Number of observation symbols
- $\boldsymbol{A} = [\boldsymbol{a_{ij}}]$: ***N*** by ***N*** state transition probability matrix
- $\boldsymbol{B} = \boldsymbol{b_j(m)}$: ***N*** by ***M*** observation probability matrix
- $\boldsymbol{\Pi} = [\boldsymbol{\pi_i}]$: ***N*** by 1 initial state probability vector

λ = (A, B, Π), parameter set of HMM

Three Basic Problems of HMMs

1. Evaluation: Given $\boldsymbol{\lambda}$, and $\boldsymbol{O}$, calculate $\boldsymbol{P}$ ($\boldsymbol{O}$ | $\boldsymbol{\lambda}$)
2. State sequence: Given $\boldsymbol{\lambda}$, and $\boldsymbol{O}$, find $\boldsymbol{Q^*}$ such that

$$P\,(Q^* \mid O, \lambda\,) = \max_Q P\,(Q \mid O\,, \lambda\,)$$

3. ***Learning: Given*** $\mathcal{X}=\{O^k\}_k$***, find*** λ^* ***such that***

$$P\,(\,\mathcal{X} \mid \lambda^*\,) = \max_\lambda P\,(\,\mathcal{X} \mid \lambda\,)$$

(Rabiner, 1989)

Calculation of ***P*** **(*O* | *λ*)**

- Marginalize over all state sequences of length T

$$P(O \mid \lambda) = \sum_{allQ} P(O, Q \mid \lambda) = \sum_{allQ} P(O \mid Q, \lambda) P(Q \mid \lambda)$$

$$P(O \mid Q, \lambda) = b_{q_1}(O_1) b_{q_2}(O_2) \ldots\ldots\ldots b_{q_T}(O_T)$$

$$P(Q \mid \lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \ldots\ldots\ldots a_{q_{T-1} q_T}$$

$$P(O \mid Q, \lambda) P(Q \mid \lambda) = \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \ldots\ldots\ldots a_{q_{T-1} q_T} . b_{q_T}(O_T)$$

- Complexity: $\boldsymbol{N^T}$

Efficient calculation of ***P*** (***O*** | ***λ***)

- Marginalize over last state:

$$P(O \mid \lambda) = \sum_j P(O, q_T = \mathrm{S_j} \mid \lambda) = \sum_j \alpha_T(j)$$

- Left to compute $\alpha_T(j)$

Calculation of ***P*** (***O*** | ***λ***)

$$\alpha_T(j) = P(O_1....O_T, q_T = S_j) = P(O_1....O_T \mid q_T = S_j)P(q_T = S_j)$$

$$= P(O_1....O_{T-1} \mid q_T = S_j)P(O_T \mid q_T = S_j)P(q_T = S_j)$$

$$= P(O_1....O_{T-1}, q_T = S_j)P(O_T \mid q_T = S_j)$$

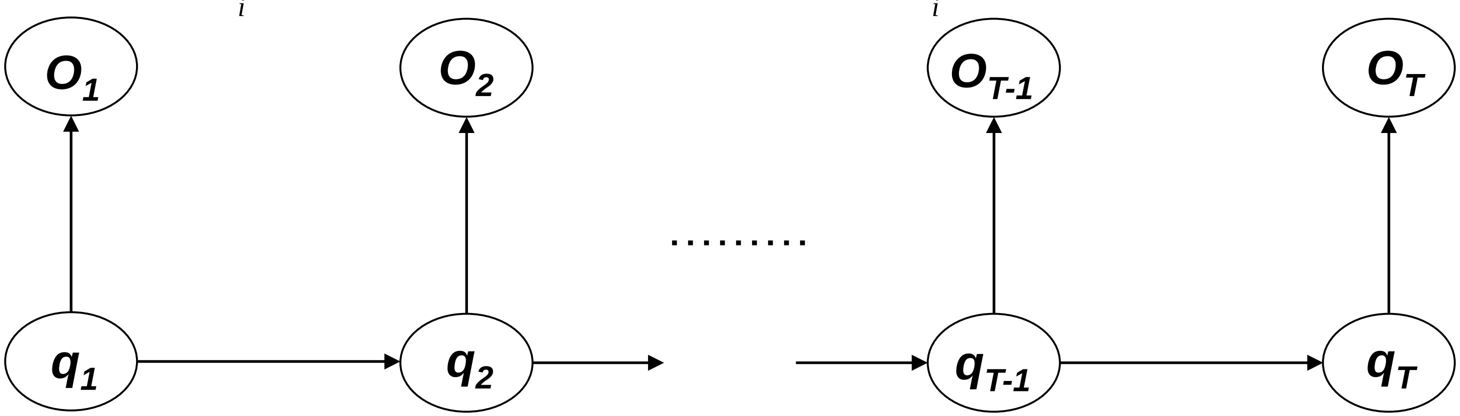
$$= \sum_i P(O_1....O_{T-1}, q_T = S_j, q_{T-1} = S_i)b_j(O_T)$$

$$= \sum_i P(O_1....O_{T-1}, q_{T-1} = S_i, q_T = S_j)b_j(O_T)$$

$$= \sum_i P(O_1....O_{T-1}, q_T = S_j \mid q_{T-1} = S_i)P(q_{T-1} = S_i)b_j(O_T)$$

$$= \sum_i P(O_1....O_{T-1} \mid q_{T-1} = S_i)P(q_{T-1} = S_i)P(q_T = S_j \mid q_{T-1} = S_i)b_j(O_T)$$

$$= \sum_i P(O_1....O_{T-1}, q_{T-1} = S_i)a_{ij}b_j(O_T) = \sum_i \alpha_{T-1}(i)a_{ij}b_j(O_T)$$



Calculation of ***P*** (***O*** | ***λ***)

- Forward variable:

$$\alpha_t(i) \equiv P(O_1 \cdots O_t, q_t = S_i \mid \lambda)$$

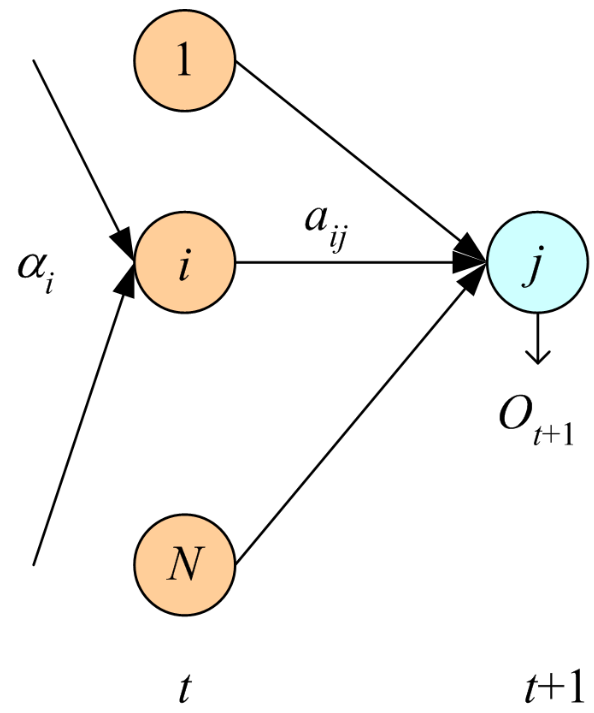
Initializa tion :

$$\alpha_1(i) = \pi_i b_i(O_1)$$

Recursion :

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

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



- Complexity?

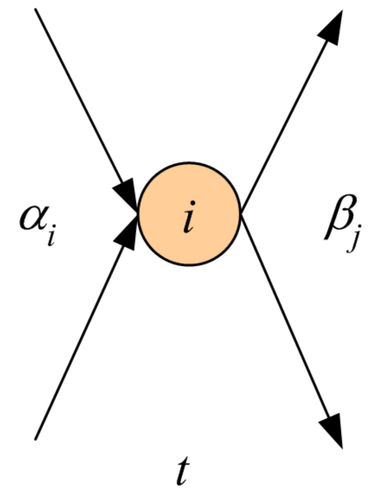
Finding the State Sequence

$$\gamma_t(i) \equiv P(q_t = S_i \mid O) = \frac{P(O \mid q_t = S_i)P(q_t = S_i)}{P(O)}$$

$$= \frac{P(O_1....O_t \mid q_t = S_i)P(O_{t+1}....O_T \mid q_t = S_i)P(q_t = S_i)}{P(O)}$$

$$= \frac{P(O_1....O_t , q_t = S_i)\underbrace{\boxed{P(O_{t+1}....O_T \mid q_t = S_i)}}_{\beta_t(i)}}{\sum_j P(O \mid q_t = S_j)P(S_j)}$$

$$= \frac{\alpha_t(i)\beta_t(i)}{\underbrace{\sum_j \alpha_t(j)\beta_t(j)}_{\text{Normalization factor}}}$$



Finding the State Sequence

Backward variable:

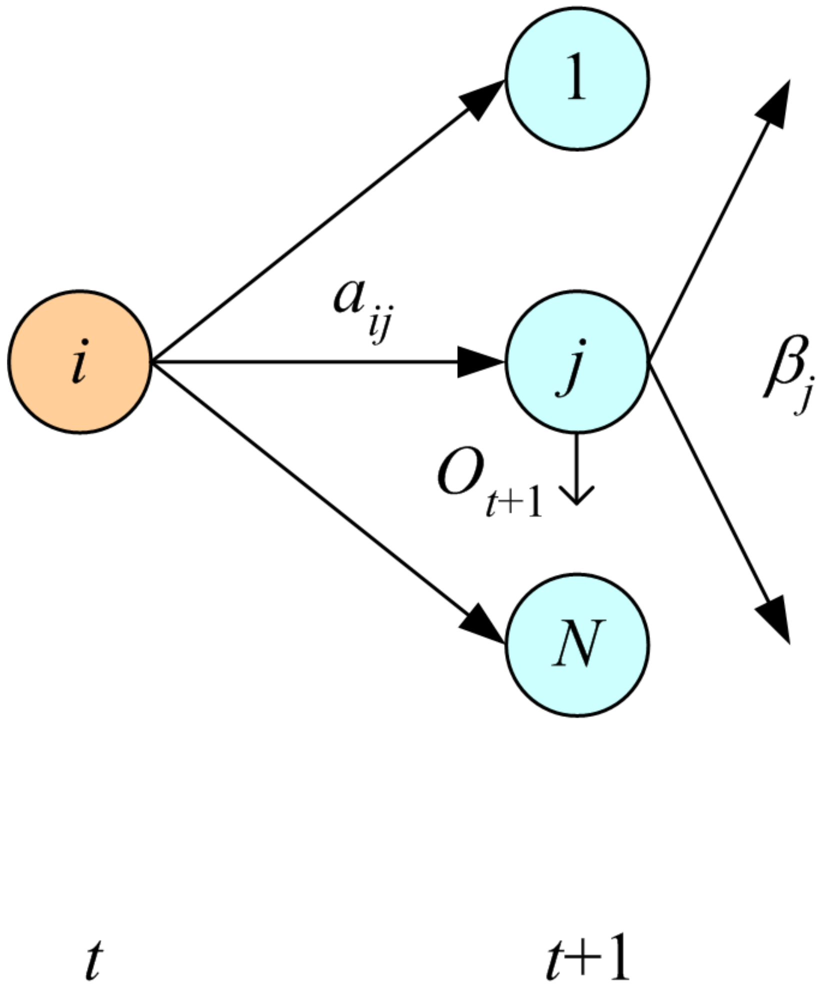
$$\beta_t(i) \equiv P(O_{t+1} \cdots O_T \mid q_t = S_i, \lambda)$$

Initialization :

$$\beta_T(i) = 1$$

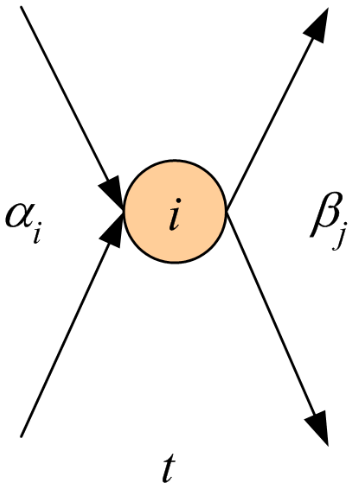
Recursion :

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



Finding the State Sequence

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}$$



Choose the state that has the highest probability, for each time step*:*

$$q_t^* = \arg\max_i \gamma_t(i)$$

No!

Viterbi's Algorithm

$$\delta_t(i) \equiv \max_{q_1 q_2 \cdots q_{t-1}} p(q_1 q_2 \cdots q_{t-1}, q_t = S_i, O_1 \cdots O_t \mid \lambda)$$

- Initialization:

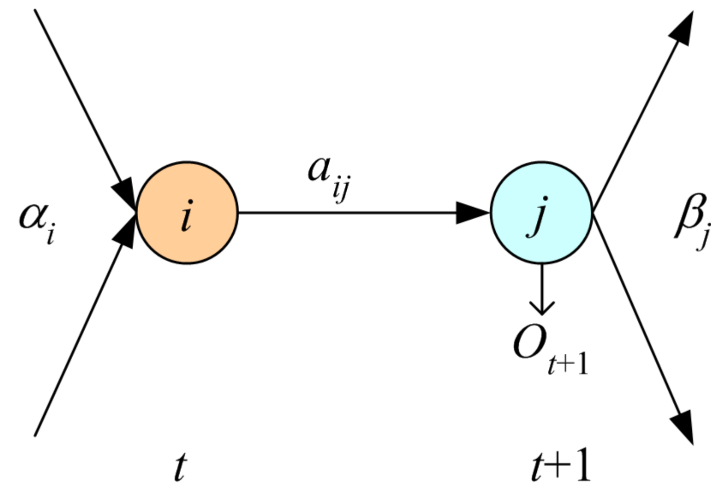
  $\delta_1(i) = \pi_i b_i(O_1)$, $\psi_1(i) = 0$
- Recursion:

  $\delta_t(j) = \max_i \delta_{t-1}(i) a_{ij} b_j(O_t)$, $\psi_t(j) = \text{argmax}_i \, \delta_{t-1}(i) a_{ij}$
- Termination:

  $p^* = \max_i \delta_T(i)$, $q_T^* = \text{argmax}_i \, \delta_T(i)$
- ***Path backtracking:***

  $q_{t-1}^* = \psi_t(q_t^*)$, $t = T, T-1, \ldots, 1$

Learning



$$\xi_t(i,j) \equiv P(q_t = S_i, q_{t+1} = S_j \mid O, \lambda)$$

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_k \sum_l \alpha_t(k) a_{kl} b_l(O_{t+1}) \beta_{t+1}(l)}$$

Baum - Welch algorithm (EM) :

$$z_i^t = \begin{cases} 1 & \text{if } q_t = S_i \\ 0 & \text{otherwise} \end{cases} \qquad z_{ij}^t = \begin{cases} 1 & \text{if } q_t = S_i \text{ and } q_{t+1} = S_j \\ 0 & \text{otherwise} \end{cases}$$

Baum-Welch (EM)

$$\text{E}-\text{step}: E\left[z_i^t\right]=\gamma_t(i) \quad E\left[z_{ij}^t\right]=\xi_t(i,j),\ \gamma_t(i)=\sum_j \xi_t(i,j)$$

$\text{M}-\text{step}:$

$$\hat{\pi}_i=\frac{\sum_{k=1}^{K}\gamma_1^k(i)}{K} \qquad \hat{a}_{ij}=\frac{\sum_{k=1}^{K}\sum_{t=1}^{T_k-1}\xi_t^k(i,j)}{\sum_{k=1}^{K}\sum_{t=1}^{T_k-1}\gamma_t^k(i)}$$

$$\hat{b}_j(m)=\frac{\sum_{k=1}^{K}\sum_{t=1}^{T_k-1}\gamma_t^k(j)1\left(O_t^k=v_m\right)}{\sum_{k=1}^{K}\sum_{t=1}^{T_k-1}\gamma_t^k(i)}$$

Continuous Observations

- Discrete:

$$P(O_t \mid q_t = S_j, \lambda) = \prod_{m=1}^{M} b_j(m)^{r_m^t} \qquad r_m^t = \begin{cases} 1 & \text{if } O_t = v_m \\ 0 & \text{otherwise} \end{cases}$$

- Gaussian mixture (Discretize using ***k***-means):

$$P(O_t \mid q_t = S_j, \lambda) = \sum_{l=1}^{L} P(\mathrm{G}_{jl})\, p(O_t \mid q_t = S_j, \mathrm{G}_l, \lambda)$$

$$\sim \mathrm{N}(\mu_l, \Sigma_l)$$

- Continuous:

$$P(O_t \mid q_t = S_j, \lambda) \sim \mathcal{N}(\mu_j, \sigma_j^2)$$

Use EM to learn parameters, e.g., $\hat{\mu}_j = \dfrac{\sum_t \gamma_t(j) O_t}{\sum_t \gamma_t(j)}$

HMM with Input

- Input-dependent observations:

$$P\left(O_t \mid q_t = S_j, x^t, \lambda\right) \sim \mathcal{N}\left(g_j\left(x^t \mid \theta_j\right), \sigma_j^2\right)$$

- Input-dependent transitions (Meila and Jordan, 1996; Bengio and Frasconi, 1996):

$$P\left(q_{t+1} = S_j \mid q_t = S_i, x^t\right)$$

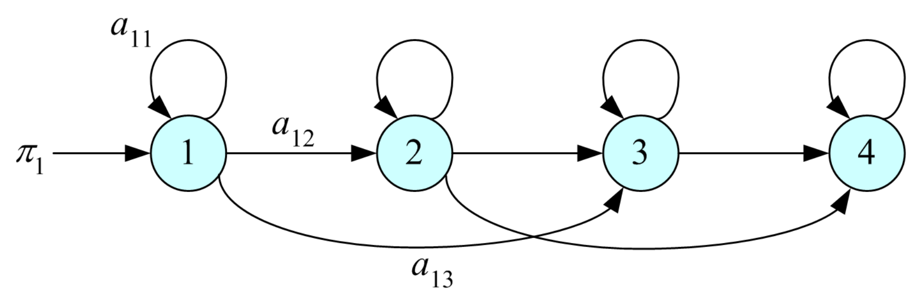
- Time-delay input:

$$\boldsymbol{x}^t = \boldsymbol{f}\left(O_{t-\tau}, \ldots, O_{t-1}\right)$$

Model Selection in HMM

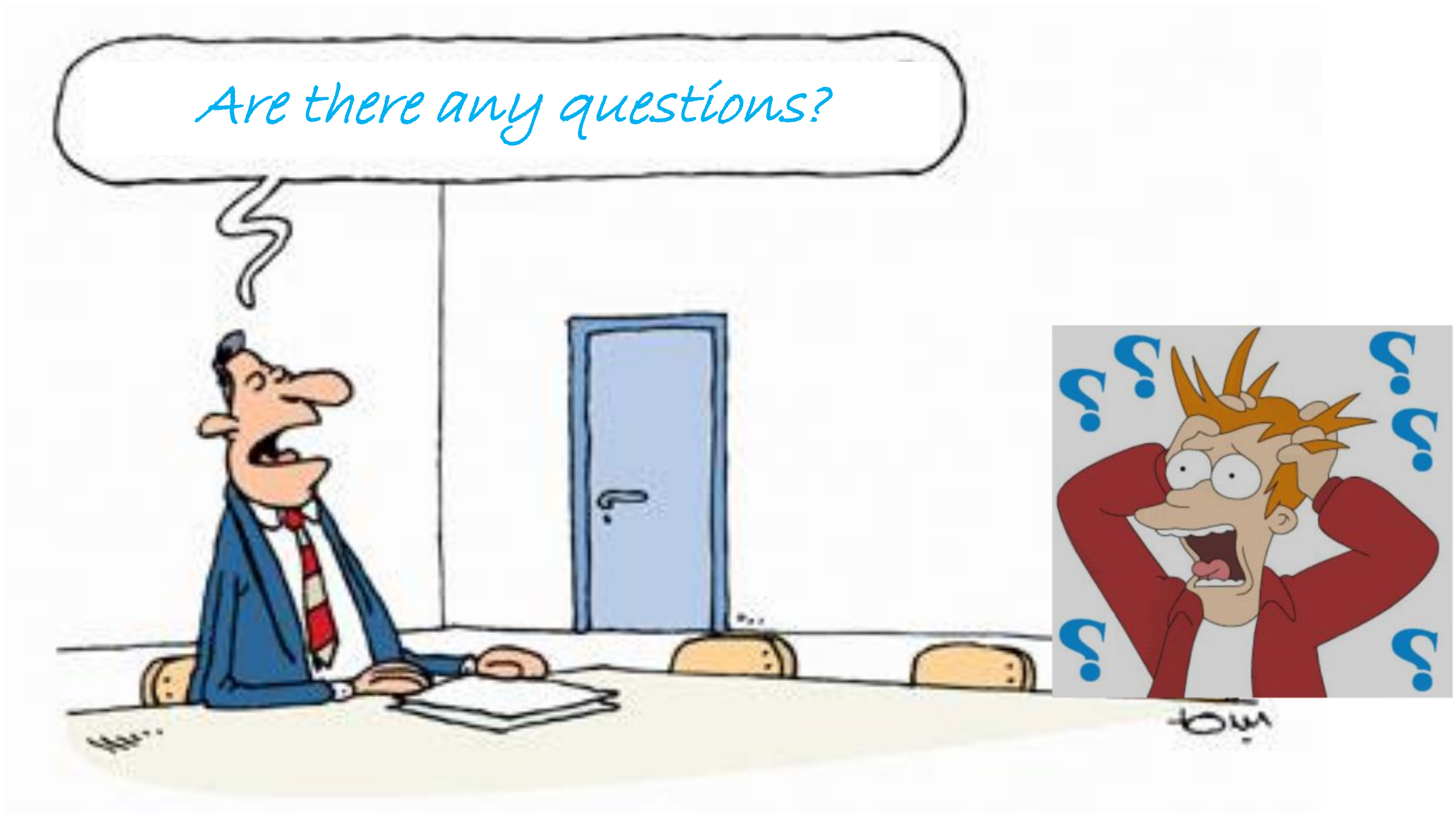
- Left-to-right HMMs:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}$$



- In classification, for each C_i, estimate $\boldsymbol{P}\,(\boldsymbol{O} \mid \lambda_i)$ by a separate HMM and use Bayes' rule

$$P(\lambda_i \mid O) = \frac{P(O \mid \lambda_i)P(\lambda_i)}{\sum_j P(O \mid \lambda_j)P(\lambda_j)}$$


Are there any questions?